Unsupervised Object Learning

Laurynas Karazija

Kellogg College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Hilary 2025

Abstract

The visual world consists of discrete, meaningful objects, which humans effortlessly perceive and segment without supervision. Emulating this ability on machines is a fundamental problem in computer vision, offering a more cognitively plausible and scalable alternative to supervised methods. In this thesis, we explore principled methods to uncover visual objects without relying on dense mask annotations. Firstly, we explore the principle of compositionality, which posits that scenes are composed of discrete, reusable objects. Numerous methods based on this principle exist, yet we note that they are limited to simplistic environments. We introduce a series of new benchmark datasets to analyse whether the current methods can scale to visually complex inputs. Most formulations do not handle the complexity of scenes well, requiring simpler uniform appearances to produce a good segmentation. Secondly, we explore the principle of common fate, which posits that entities that move together should be grouped together. We design several loss functions that connect mask predictions with estimates of scene motion to handle binary and multi-object scenarios. Our proposed formulations can be applied to various existing segmentation methods, complementing their learning principles with learning from motion. We then consider the limitations of instantaneous motion and propose incorporating long-term motion information using sparse point trajectories. To enable this, we design a loss function that enforces the idea that trajectories within an object should have much redundancy. Finally, we explore how existing structures in language can be used to learn object segmentation without the need for any dense mask annotations. We construct a method for open-vocabulary segmentation that uses a pre-trained text-to-image diffusion model to connect language with visual representations of objects. It avoids the need for any further training, showing how text-to-image diffusion models are also powerful open-vocabulary segmentation methods.

This thesis is submitted to the Department of Engineering Science, The University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Laurynas Karazija, April 2025.

Acknowledgement

Throughout my studies, I have been blessed with amazing people who have supported me on this journey. First and foremost, I am deeply grateful to my supervisors, Andrea Vedaldi, for the constant encouragement and deeply insightful moments on the whiteboard; Christian Rupprecht, for the clarity, level-headedness and compassion during both good and difficult times; and Iro Laina, for the understanding, motivation and help to be better. Without their guidance and patience, this would not have been possible. I would also like to thank the excellent administrative staff of AIMS, Wendy, whose constant availability, compassion and organisation have filled this journey with great conference experiences, amazing AIMS events and warm pub nights. I would also like to thank Ash, whose mastery over Triton and Athena, constant availability and helpfulness have made the work presented here possible. I wish to thank all the VGG members for the great research environment, for the insightful discussions, for the amazing days and nights pushing the deadlines, and for the amazing parties thereafter. A special shoutout goes to my travel crew: Luke, Suny, Charig, Tim, and Paul. We scaled mountains, hiked gorges, and breached beaches. I will never forget the amazing times we had. I am also deeply grateful to Luke and Noel for our long, insightful discussions on unsupervised segmentation. I met an amazing group of people at AIMS who became wonderful people to talk to and call on when we all needed a little break, wanted board games, pub or a laugh. Most importantly, I want to thank my family. Thank you for the unconditional love, unwavering support, and undoubting confidence that I have always felt from you. Thank you for always being able to talk, always listening and believing in me when I no longer could. Finally, I want to say thank you to countless others who listened, asked, and encouraged me on this journey.

Contents

Chapter 1

Introduction

While reading this line, it is effortless to distinguish the text from the background, the page from its surroundings and even objects in the peripheral view. We perceive the world not as an arrangement of retinal responses or a grid of pixel values but as a composition of distinct, coherent objects. This intuitive segmentation—knowing where one thing ends and another begins—is fundamental to seeing, reasoning, and acting. As we endeavour to construct increasingly autonomous and intelligent machines, it becomes essential to endow them with the same capability.

Computer vision has seen significant progress, enabling understanding of scene layouts [V. S. Chen et al. 2019], linking images with language [Radford et al. 2021], or even generating new imagery based on textual description [Ramesh et al. 2022] or 3D scenes from a single image [Gao et al. 2025]. Segmentation, the task of labelling each pixel with an object it belongs to, is crucial across diverse systems, from medical imaging and wildlife monitoring to self-driving, robotics, and media editing. Recently, powerful prompted segmentation models showing unprecedented accuracy and generality have emerged [Kirillov et al. 2023; Zou et al. 2023; Ravi et al. 2024]. This significant progress has been achieved thanks to the large amounts of annotated data used to *supervise* the learning process.

Reliance on extensive supervision is in stark contrast to human cognition. We can easily understand new visual concepts, extrapolate beyond situations encountered in everyday life, and reason in high-level terms. Objects are fundamental to us, often described as one of the core systems of knowledge [Spelke and Kinzler 2007]. Moreover, we receive little instruction or supervision to build our perception and

understanding of the visual world, developing principles of object representation with little experience [Spelke 1990].

In this thesis, we explore how one can learn to decompose the visual world, as seen in images and videos, into objects without relying on mask annotations. Data annotation, particularly for segmentation, is highly labour-intensive. For example, annotating Cityscapes, with only 30 classes, required 1.5 man-hours per image [Cordts et al. 2016; Shin et al. 2021]. Annotation can also introduce biases [Wilson et al. 2019; Gustafson et al. 2023], and can contain errors and inaccuracies [H. Zhang et al. 2020]. Furthermore, supervised models are constrained by fixed ontologies defined by the annotated data. Unseen objects or novel categories are not handled well, requiring careful outlier detection [Nayal et al. 2023]. Even a promptable segmentation model trained on a billion masks [Kirillov et al. 2023] shows limitations, especially with boundaries and fine structures [Osco et al. 2023], suggesting that even more annotations are needed. Overall, annotation is onerous, noisy, limiting, and even impractical when encountering entirely new objects or continuously acquiring new data.

Learning to segment the world into discrete symbolic objects without annotations is an essential area with high impact. It can alleviate data annotation issues and enable higher-level object-centric reasoning in downstream applications. Object-level reasoning, in turn, shows promise in learning physics-enabled world-models [Watters et al. 2017; Van Steenkiste et al. 2018; Z. Wu et al. 2023], improving visual reasoning [Deitke et al. 2024; J. Yang et al. 2023], enhancing RL agents [Watters et al. 2019a; Zadaianchuk et al. 2023c], and offering robustness to distribution shifts [Dittadi et al. 2022].

In this thesis, we explore the problem of segmenting objects without mask annotations through three sources of possible learning signals. First, we consider appearance only and investigate compressing and reconstructing the scene from a discrete set of vectors. Second, we combine appearance *and* motion, exploring how to train visual segmentation networks using motion observations. Finally, we use structure in natural language to help find and delineate objects in images.

sofa & bed sofa cushion & mattress

Figure 1.1: Objects form hierarchies. Both a sofa and a bed are objects. One might also call a sofa cushion or a mattress an object, but also they are part of the sofa and the bed. The requisite granularity can depend on the task and situation.

1.1 What is an Object?

As this thesis centres around objects, it is important to consider what constitutes an object. While objects' precise definition and nature have been a subject of intense ongoing debate in metaphysics—whether they exist independently of the mind perceiving them or are inseparable from it—we remain agnostic to these questions.

Here, an object is a visually perceivable entity: something a camera can record. We further require that the objects have a discernible extent within the image or frame. This property enables using segmentation to measure how well a model has learned visual objects. Segmentation is often useful output by itself. If there is no representation or encoding of each visual object in the scene, we can construct one as we know which pixels belong to which object [Hénaff et al. 2021].

Rather than a metaphysical question of objecthood, a more pragmatic question is that of hierarchical granularity. Consider a room shown in fig. 1.1 with the sofa and the bed. Resting on top of these objects are a sofa cushion and a mattress. Each is part of the sofa or bed, respectively, yet the cushion and the mattress could be objects in their own right. A person might flip the cushion or adjust the mattress—perceiving, reasoning and acting on them as separate objects. Objects form hierarchies through part-whole relationships. Should systems consider all potential hierarchy levels, i.e. both the sofa and the cushion in the example above? This would be unnecessary for a house-moving robot. The move could be efficiently planned by considering pieces of furniture as single objects—moving the sofa would

also move the cushion. However, a cleaner robot might take extra care to steam both sides of the cushions and the mattress. The granularity of what an object is might be task, situation, or even context-dependent.

A straightforward solution exists in the supervised case: annotation masks are provided at the required granularity, ensuring models are useful for intended tasks. As annotations are often crowd-sourced, they tend to be intuitive and broadly sensible. The annotations are both a learning signal and a definition of an object. When annotations are unavailable, one must define a principle for grouping pixels to form an optimisation objective; a learning signal is still required. In this thesis, we reflect on the supervised case and consider that different principles for learning objects will also somewhat define what those objects are. For example, using interaction (e.g., lifting) biases learning towards manipulable entities, potentially omitting others. Thus, a *good* learning signal should accommodate a wide range of objects. As we evaluate whether objects are located and delineated based on ground truth masks, disagreements between labelled objects and those found by the model will also emerge. Thus, in this thesis, objects have two imperfect definitions. The first is human-aligned but implicit—occasionally revealed only through example annotations when evaluating. It reflects shared intuition but is never fully specified nor widely available. The second arises from the learning signal itself. The goal is to find and implement principles for learning to segment objects such that the gap is minimal.

1.2 Literature Review and Key Ideas

We now explore different sources of learning signals for segmenting objects. We first explain different segmentation paradigms. We then review the relevant literature and explore the key ideas. We highlight several principles for learning to segment objects. We note that the principles are not rules but useful heuristics that facilitate segmentation.

Segmentation paradigms. Several formulations of the segmentation paradigm exist. At the simpler end, one can consider binary segmentation, often phrased as saliency prediction or foreground-background segmentation. Here, one seeks to find a main subject or subjects captured in the image that is often well-framed

and contains only one or a few objects. Semantic segmentation is a more general formulation that classifies each pixel into one of some known categories without discerning between individual objects of similar type, e.g., producing a single collective mask for all the dogs in an image. In the unsupervised case, learnable centroids replace predefined categories. Instance segmentation generalises semantic segmentation by requiring each object in the class of interest to be assigned an individual mask. Notably, instance segmentation masks might overlap—the same pixel might be assigned to multiple objects because masks are annotated independently. Some methods perform closely related multi-object segmentation, which individuates objects but, like semantic segmentation, assigns each pixel to exactly one component. To account for uncountable or uninteresting "stuff" classes (e.g., water, sky, foliage), panoptic segmentation [Kirillov et al. 2019] combines instance and semantic segmentation to jointly predict "things" as instances and "stuff" as regions.

1.2.1 Learning from Appearance

As we concentrate on visual objects, it is natural to start by considering the visual appearance of an object, that is, learning from the raw input pixel values directly. While having the least amount of input, principles for learning from only appearance are the most foundational, as they can be applied to other modalities or combined with other principles.

To segment an object, one needs to group appropriate pixels. The question is how to determine which pixels should be grouped. Gestalt psychology [Wertheimer 1912] suggests principles of grouping that follow continuity, similarity and proximity. Early approaches relied primarily on hand-crafted features and statistical methods facilitated by such ideas to group pixels based on colour and texture similarity and spatial proximity. Classical approaches included normalised cuts [Jianbo Shi and Malik 2000], mean shift clustering [Comaniciu and Meer 1999], and SLIC superpixels [**achanta2010slic**] amongst others. Building on top of these, hierarchies of segments could also be induced [Arbelaez et al. 2010; Arbelaez et al. 2014]. While interpretable and computationally efficient, these approaches often struggled with complex scenes and higher-level semantic grouping, falling out of favour in the deep learning era.

Self-supervised learning. Instead of grouping pixel values directly, one can use a neural network to extract features. Training networks to provide such features is a matter of self-supervised representation learning, where models learn from the input data by solving a "pretext task". Early methods used tasks like colourisation, predicting rotation and jigsaw solving [Doersch et al. 2015; Doersch and Andrew Zisserman 2017; Gidaris et al. 2018; Richard Zhang et al. 2016]. The intuition is that by learning to perform such artificial tasks, the networks must learn image semantics, such as orientations of objects, colours, compositions, *etc.*, capturing that information in the features useful for downstream tasks.

Instance discrimination is a widely used pretext task where representations of images that should be similar are pushed together while (optionally) pushing away from dissimilar examples [Grill et al. 2020; Caron et al. 2021; Caron et al. 2020; K. He et al. 2020]. The notion of similarity is often artificially induced by augmenting the data with transformations such as cropping or applying photometric distortions. Similar ideas can be applied densely as well, on object, patch, or pixel level [Ji et al. 2019a; Cho et al. 2021; Hénaff et al. 2022; Xinlong Wang et al. 2021; Van Gansbeke et al. 2021; Ziegler and Asano 2022; X. Zhang and Maire 2020]. For object level, e.g., [X. Zhang and Maire 2020; Hénaff et al. 2022], this would mean making pixels features within an object similar. A related pretext task is based on input reconstruction that has been masked or corrupted, requiring the model to recognise and use relevant context to predict the missing parts [K. He et al. 2022; Vincent et al. 2008; Pathak et al. 2016; J. Zhou et al. 2021]. In the case of masking [K. He et al. 2022], the model might look at visible patches of objects to infer their presence and reconstruct the rest. This suggests the following principle:

Consistency: objects should be internally consistent, allowing parts to be inferred from and inform each other. P1

Notably, the patch representations learned with DINO [Caron et al. 2021] on visual transformers [Alexey Dosovitskiy et al. 2021] contain excellent semantic object descriptors [Amir et al. 2021] that do not change a lot within the object. Segmentation of objects can be formed by simply clustering such powerful features [Melas-Kyriazi et al. 2022a; Y. Wang et al. 2022; Siméoni et al. 2021; Siméoni et al. 2023; Shin et al. 2022a] reverting to principles of connectedness, similarity and proximity but now in the learned space. DINO features have transformed the unsupervised

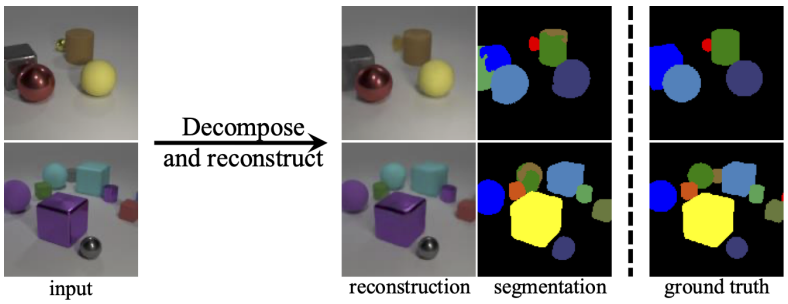


Figure 1.2: Output of a trained compositional reconstruction model [Locatello et al. 2020]. Compositional approaches predict segments that are recomposed to reproduce the input.

segmentation approaches, enabling the construction of powerful saliency prediction models. Alternatively, one can distil the alignment of learned features, clustering them over a dataset to form unsupervised semantic segmentation models [Hamilton et al. 2022; Seong et al. 2023; C. Kim et al. 2024].

Unsupervised instance segmentation approaches build upon simpler segmentation techniques to recognise many objects in the scene. Unsupervised saliency predictors [Melas-Kyriazi et al. 2022a; Y. Wang et al. 2022; Siméoni et al. 2021], e.g., based on self-supervised features, are first used to pseudo-label object-centric datasets with masks. Then, a segmentation network is supervised using these noisy predictions [Van Gansbeke et al. 2021; Xinlong Wang et al. 2022; Zadaianchuk et al. 2023a; Xudong Wang et al. 2023a; D. Li and Shin 2024; Arica et al. 2024]. A similar idea has been previously used in scaling saliency prediction [J. Zhang et al. 2018; Y. Zeng et al. 2019; Nguyen et al. 2019], where hand-crafted methods served as a starting point. The key intuition is that object segmentation is cold-started using a different, weaker method. The task is then rephrased as learning from noisy and incomplete labels, where, due to automated labelling, abundant data is available.

Compositional reconstruction. Approaches like AIR [S. M. A. Eslami et al. 2016], SPAIR [Crawford and Pineau 2019], and GNM [Jiang and Ahn 2020] describe a structured generative model of images. The images are assumed to be composed by sampling multiple individual objects and placing them within the scene. As the true posterior of this process is often intractable, these methods approximate it with recognition models for identifying objects, their extent and location. Alternatively,

discriminative approaches [C. P. Burgess et al. 2019; Locatello et al. 2020; Monnier et al. 2021] attribute discrete parts of the scene to different *slots*—a collection of variables that represent an object. These are very similar in spirit and, at a high level, differ mostly by the degree to which the probabilistic framework is employed. Collectively, these methods are based on *compositional* reconstruction [Emami et al. 2021; Z. Lin et al. 2020b; Engelcke et al. 2020; Engelcke et al. 2021; Smirnov et al. 2021]. These methods form a segmentation for the purpose of assigning or *binding* areas of the image to one of the available compressed representation slots [Greff et al. 2020]. The representations are used to recompose the image back from this spatial mixture (see fig. 1.2 for an example output). The proposition is that the following hypothesis holds [C. P. Burgess et al. 2019]:

Composability: the need to recompose the scene from discrete compressed slots will lead to identifying simple and reusable entities, i.e. objects. P2

1.2.2 Learning from Motion

Reacting to changes in the environment is important for survival. Our attention is naturally drawn to the motions we perceive. Goodale and Milner [Goodale and Milner 1992] describe that the human visual system supports a separate pathway for perceiving motions in addition to the one for appearance. In addition to focusing our attention, motion provides a powerful perceptual grouping principle. Gestalt psychologists described this as the principle of common fate [Wertheimer 1912]: visual stimuli that move together are perceived as belonging to the same whole object. The principle forms a very strong basis for early object perception in infants [Spelke 1990]. It has also been a source of inspiration in computer vision. The basic intuition is that

Common Fate: pixels of an object will move together and largely independently of the rest of the scene. P3

This enables grouping them by motion into a single object.

Representing motion. To leverage motion for learning objects, one must first be able to extract and represent it for a given video. Most commonly, motion is represented as optical flow [Gibson 1950]—a vector field that describes the position change of each pixel in the image. Other representations include tracking

Figure 1.3: Video frame and associated optical flow field (colorised). The motion makes the bear pop out from the surroundings. Image from the DAVIS dataset [Perazzi et al. 2016]. Flow estimated using RAFT [Teed and J. Deng 2020].

keypoints [Tomasi and Kanade 1991] or points [Sand and Teller 2008]. Optical flow can be solved for by considering local matching of pixel intensities subject to various constraints [Lucas and Kanade 1981; Horn and Schunck 1981]. Alternatively, one can *learn* to estimate the flow from data in supervised [A. Dosovitskiy et al. 2015; Ilg et al. 2017; D. Sun et al. 2018; Teed and J. Deng 2020] or self-supervised [Liang Liu et al. 2020; Stone et al. 2021; H.-P. Huang et al. 2023] manner. Modern motion estimation methods are trained on purely synthetic datasets yet are able to generalise to real-world scenarios. For example, point trackers [Doersch et al. 2022; Karaev et al. 2024; Doersch et al. 2023] learn from simulation data with randomly sampled scenes supervised with a physics and a rendering engine [Greff et al. 2022]. This enables the estimating and leveraging of such modalities to localise objects, which is typically done in motion or video object segmentation tasks.

Motion segmentation has evolved from early methods that consider moving layers [Jepson and Black 1993], probabilistic mixtures of motion models [Torr 1998], to clustering point trajectories over time to identify moving regions [Brox and Malik 2010b], and leveraging motion boundaries [Papazoglou and Ferrari 2013]. Optical flow, having similar dimensions as an image, is a popular choice for injecting motion into segmentation networks [Tokmakov et al. 2017; Yanchao Yang et al. 2019]. The main intuition is that the optical flow field within an object is coherent, changing smoothly, yet often very different from the background (see fig. 1.3). This causes the objects that move to appear distinct from their surroundings. Motion is a simpler modality than appearance, without difficult factors like lighting or texture. Due to such simplicity, some works focus entirely on motion [Charig Yang et al. 2021; Meunier et al. 2022; Meunier and Bouthemy 2023a].

Video-only methods. Some works have also considered learning from video without explicitly representing motion information. Time dimension offers additional opportunities to construct pretext tasks that challenge temporal coherence understanding by, e.g., ordering frames or clips [Fernando et al. 2017; H.-Y. Lee et al. 2017; Misra et al. 2016; D. Wei et al. 2018]. Alternatively, one can try to model the temporal evolution of the scene [Jabri et al. 2020; Salehi et al. 2023; Xiaolong Wang et al. 2019; Han et al. 2019]. For segmentation, compositional reconstruction (P2) models are extended to allow for slots to propagate or bind in the time dimension [Kosiorek et al. 2018; Jiang et al. 2020; Kabra et al. 2021; Zoran et al. 2021; Weis et al. 2021; Zadaianchuk et al. 2024; Aydemir et al. 2024].

1.2.3 Learning from Language

Grouping of visual stimuli based on spatial or spatiotemporal patterns is critical for learning to perceive objects. We have reviewed several principles enabling learning to discern visual objects bottom-up. What if one had some signal that already encoded some structure in the world? Could we use that for learning in a top-down manner?

Classes. Image classification is a longstanding problem in computer vision, leading to the creation of large-scale datasets with classes annotated for each image [Russakovsky et al. 2015]. These annotations can provide a top-down signal of what to look for. One approach to exploiting this information is employing a trained convolutional classifier to localise the object. This can be achieved via backpropagation [Simonyan et al. 2013] or by examining the final aggregation layer in the network to form class activation maps [Oquab et al. 2015; B. Zhou et al. 2016]. Such maps can seed a localisation method, which can be further refined with training [Kolesnikov and Lampert 2016; Jing et al. 2019]. However, the class labels are required for the images, which, while not as laborious as dense annotations, are limited to predefined classes and rarely available.

Naming. Language is a standout feature of human intelligence. The language is already divided into meaningful concepts and units. It is the medium through which we can communicate and reason about the world, referring to different objects with discrete terms. Learning from this existing structure can be a powerful top-down

signal for segmenting objects. We assume the objects are already assigned *names* that get mentioned in texts associated with images. The *names* represent objects, at least in language. These descriptors are often highly compact, encoding only the most essential aspects of the objects. In large image-text collections, the same names will co-occur with similar objects. Thus, the idea for learning segmentation is the following principle:

Naming: connect visual objects through shared *names* to learn to identify their extent. P4

Next, we review how to connect images to text and text to objects.

Vision-language models. Practically, obtaining a large-scale collection of paired image-text data is nearly as cheap as obtaining just images. For example, this can be achieved by crawling the Internet [Sharma et al. 2018; Schuhmann et al. 2022] and considering alt-text. The scale and availability of such paired data have ushered in a wave of Vision-Language Models (VLMs). The main intuition behind common VLMs is to learn two networks, one for images and one for text, using contrastive loss between their outputs [Radford et al. 2021; Jia et al. 2021; Zhai et al. 2023]. This way, one can construct representations of the images and text aligned in a shared feature space. In addition to contrastive objectives, other schemes like masked modelling, captioning, and text generation are also used [A. Singh et al. 2022; J. Li et al. 2023; X. Chen et al. 2023]. Most VLM approaches offer only global image descriptors, as the data is only available at the image level.

Open-vocabulary segmentation. VLMs signalled a shift from a closed world of predefined categories to an open world of arbitrary classes described in language. Segmenting objects described in the language is a task of open-vocabulary segmentation [Bucher et al. 2019; P. Li et al. 2020; Gu et al. 2020; J. Cheng et al. 2021], where a set of target categories is provided as text, and the goal is to assign each pixel to one of these classes.

One can solve this problem in a supervised manner by using, for example, annotated segmentation masks and text labels associated with them. However, obtaining mask annotations at scale remains a problem.

Approaching this problem without masks involves constructing features for image

regions and aligning them with corresponding text embeddings. This can be done by modifying how information is aggregated in the image encoder to align local features [C. Zhou et al. 2022; Cha et al. 2023; Ranasinghe et al. 2023]. Visual objects' textual identifiers are often noun-like entities, which can be identified in the text by part-of-speech tagging [Brill 1992]. Thus, another useful idea is to consider candidate objects by extracting noun phrases from captions and aligning them with perceptual groups found in the image [Jiarui Xu et al. 2022; Jilan Xu et al. 2023; Q. Liu et al. 2022]. Most of these approaches carry a significant computation cost, as large-scale training or fine-tuning of the model is required.

1.3 Outline and Contributions

This thesis explores three different sources of signals for learning objects. We first explore learning from appearance using the principle of *composition* (P2). We then exploit the principle of *common fate* (P3) to learn to segment objects from motion. Finally, we consider language by exploiting *naming* (P4). In this section, we outline each theme of the thesis and its constituent chapters, as well as the contributions of each chapter.

Appearance

At first, we explore learning to segment objects from appearance alone. We concentrate on the compositional scene appearance reconstruction (P2) as a principled mechanism for decomposing scenes into objects. Many recent approaches to unsupervised segmentation that explore such principles are trained in visually simple environments. We ask whether these methods can scale to visually complex scenes. In chapter 2, we propose a range of new benchmark datasets of increasing visual complexity by introducing rich material, texture and shape details. We concentrate on progressively increasing visual fidelity in a simulation to provide a middle ground between simple settings used before and highly intricate real-world scenes. In addition to new challenging datasets, we advocate for the use of metrics that capture many facets of segmenting objects. We propose to pay attention to whether objects are separated from the background with appropriate boundaries and consider the relative size of objects. We survey and test a range of state-of-the-art (SOTA)

methods on the new datasets, comparing the effects of their formulations and highlighting the limitations of current art.

Motion

In this part, we start by exploring how motion can be used to learn binary object segmentation. Drawing from the principle of common fate (P3), we seek to train segmentation networks to predict regions that will move together. Our intuition is to learn object segmentation in the visual domain for generality but exploit motion as a source of a learning signal. In chapter 3, we propose a guess-and-check approach and a loss function for training an RGB segmentation network using optical flow instead of annotations. A segmentation model must predict an object that *might* be moving. The loss then checks if the predicted object could explain the motion observed in the optical flow. The check is performed by using a least-squares solution of a parametric motion model. One can also view this as applying the consistency principle (P1) to flow vectors—we explain all flow pixels within an object with a shared model.

In chapter 4, we extend this paradigm to a multi-object setting by replacing the *min-out* of the motion model with a *prior*. We propose a novel loss that measures the likelihood of an optical flow region carved out by predicted masks. The intuition is that the motion can approximately be modelled as a coordinate transformation, and we can consider a prior distribution of possible transformations and marginalise them. We show how this principled approach can deal with multi-object settings and even be applied to real-world videos. We also show that by implementing this idea as a loss, we are able to augment existing appearance-based approaches to learn from motion.

Not all objects are necessarily in motion at all times, and objects might move together for a long time, suggesting that highly informative motion is sparse. In chapter 5, we explore how one could model and learn from long-term motion observation, such as point tracks. While the evolution of object motion is tricky to model, we posit that the motion of points within objects must be self-consistent: the motion of points within an object could be described by a few trajectories (P1). Using this idea, we propose a principled loss function that can train a segmentation network using point tracks as a learning signal. The intuition is to predict image

areas where trajectory grouping contains a lot of redundancy.

Language

In this part, we explore how vision-language models can be used to construct image segmenters. The central idea is to leverage the structure already present in language, connecting textual names with visual objects (P4). As the use of language enables one to be specific about segmentation targets, we explore an open-vocabulary segmentation task. In chapter 6, we use the knowledge in the pre-trained large-scale generative text-to-image diffusion networks to produce a segmentation model. We introduce a method for approaching this task that departs significantly from existing art in building a segmentation model without access to mask annotations. The central idea is to exploit a language-conditioned image generator to bridge the vision-language gap and attribute generated image areas to different words by exploiting the conditioning mechanism. We use this to produce visual descriptions we call prototypes, enabling us to perform segmentation by comparison with image features.

1.4 Publications

Chapters 2 to 6 each contain a research paper that has been peer-reviewed and published at an international conference.

Chapter 2: "CLEVRTEX: A Texture-Rich Benchmark for Unsupervised Multi-Object Segmentation" **Laurynas Karazija**, Iro Laina, Christian Rupprecht. *In Conference on Neural Information Processing Systems (NeurIPS) 2021 Datasets and Benchmarks Track*.

Chapter 3: "Guess What Moves: Unsupervised Video and Image Segmentation by Anticipating Motion" Subhabrata Choudhury*, **Laurynas Karazija***, Iro Laina, Andrea Vedaldi, Christian Rupprecht. *In British Machine Vision Conference (BMVC), 2022*. (Spotlight)

Chapter 4: "Unsupervised Multi-object Segmentation by Predicting Probable Motion Patterns" **Laurynas Karazija***, Subhabrata Choudhury*, Iro Laina, Christian Rupprecht, Andrea Vedaldi. *In Advances in Neural Information Processing Systems*

(NeurIPS), 2022.

Chapter 5: "Learning segmentation from point trajectories" **Laurynas Karazija**, Iro Laina, Christian Rupprecht, Andrea Vedaldi. *In Advances in Neural Information Processing Systems (NeurIPS), 2024.* (Spotlight)

Chapter 6: "Diffusion Models for Open-Vocabulary Segmentation" **Laurynas Karazija**, Iro Laina, Andrea Vedaldi, Christian Rupprecht. *In European Conference on Computer Vision (ECCV), 2024.* (Oral)

* indicates equal contribution.

Chapter 2

ClevrTex: A Texture-Rich Benchmark for Unsupervised Multi-Object Segmentation

This chapter presents a paper that has been published in *Conference on Neural Information Processing Systems (NeurIPS)*, 2021, *Datasets and Benchmarks Track*.

In this chapter, we explore methods that learn from the appearance of the scenes, meaning the RGB images or videos themselves. We introduce a series of benchmarks called ClevrTex that are designed to evaluate the performance of multi-object segmentation models in an effort to scale them to real-world settings. We also benchmark the performance of state-of-the-art multi-object segmentation methods that are based on a compositional reconstruction (P2). Our experiments show that applying various materials to only objects already posed a significant challenge, as the appearances were no longer uniform. Our findings showed that the current state-of-the-art methods were limited in their ability to handle visually complex scenes. ClevrTex has since become a standard benchmark for multi-object segmentation methods, inviting the community to scale them to visually complex environments. Follow-up works greatly improved performance on ClevrTex by scaling encoders and considering more involved decoding architectures. Further works used self-supervised features to begin scaling compositional reconstruction approaches to real-world benchmarks.

CLEVRTEX: A Texture-Rich Benchmark for Unsupervised Multi-Object Segmentation

Laurynas Karazija, Iro Laina, Christian Rupprecht

Visual Geometry Group

University of Oxford

Oxford, UK

{laurynas,iro,chrisr}@robots.ox.ac.uk

Abstract

There has been a recent surge in methods that aim to decompose and segment scenes into multiple objects in an unsupervised manner, i.e., unsupervised multi-object segmentation. Performing such a task is a long-standing goal of computer vision, offering to unlock object-level reasoning without requiring dense annotations to train segmentation models. Despite significant progress, current models are developed and trained on visually simple scenes depicting mono-colored objects on plain backgrounds. The natural world, however, is visually complex with confounding aspects such as diverse textures and complicated lighting effects. In this study, we present a new benchmark called CLEVRTEX, designed as the next challenge to compare, evaluate and analyze algorithms. CLEVRTEX features synthetic scenes with diverse shapes, textures and photo-mapped materials, created using physically based rendering techniques. It includes 50k examples depicting 3-10 objects arranged on a background, created using a catalog of 60 materials, and a further test set featuring 10k images created using 25 different materials. We benchmark a large set of recent unsupervised multi-object segmentation models on CLEVRTEX and find all state-of-the-art approaches fail to learn good representations in the textured setting, despite impressive performance on simpler data. We also create variants of the CLEVRTEX dataset, controlling for different aspects of scene complexity, and probe cur-

rent approaches for individual shortcomings. Dataset and code are available at https://www.robots.ox.ac.uk/~vgg/research/clevrtex.

2.1 Introduction

Supervised scene understanding has seen significant progress in the last decade. The introduction of deep learning to the field and large, manually annotated datasets have made it possible to address tasks such as object detection [Li Liu et al. 2020], semantic or instance segmentation [K. He et al. 2017], layout prediction [D. Xu et al. 2017] and dense captioning [Johnson et al. 2016] with considerable accuracy. However, in absence of labels, and thereby supervision, such tasks are exceedingly difficult, even though it is easy to imagine that with enough images (or videos), it should be possible to identify objects and the general composition of a scene without human annotations. This renders unsupervised multi-object segmentation, as well as object-centric learning a challenging yet promising field with high potential.

While certain tasks in the general context of *unsupervised* scene understanding and decomposition have a relatively long history in computer vision, the majority of applications focus on single objects: image classification [Van Gansbeke et al. 2020; Ji et al. 2019b; Caron et al. 2018], saliency detection [J. Zhang et al. 2018; Nguyen et al. 2019], foreground/background segmentation [M. Chen et al. 2019; Bielski and Favaro 2019; Voynov et al. 2020; Melas-Kyriazi et al. 2022b] and general image-level representation learning [K. He et al. 2020; T. Chen et al. 2020; Caron et al. 2021; Grill et al. 2020]. These methods are usually developed on datasets such as ImageNet [Russakovsky et al. 2015] that contain one object of interest per image. Nevertheless, most real-world scenes are often comprised of multiple objects in varying spatial configurations.

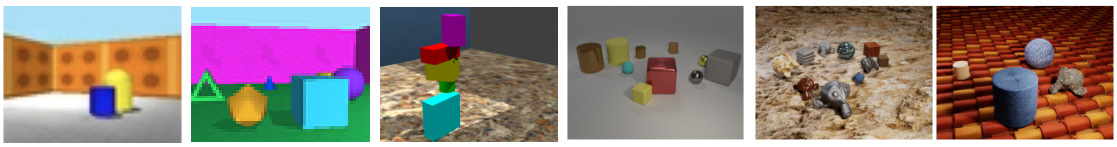
Only recently, methods have been developed to analyze and decompose whole scenes containing multiple objects, i.e., jointly learning to represent and segment objects from raw image input, *without* supervision. However, since moving from individual objects to complex scenes drastically complicates the problem, these methods currently rely on simple synthetic datasets. The complexity of these datasets ranges from simple, single-color 2D shapes arranged against a black background [C. P. Burgess et al. 2019] to rendered 3D scenes composed of uniformly

colored, 3D primitives (cubes, spheres, cylinders) [Johnson et al. 2017] (fig. 2.1). Interestingly, current methods work *very well* on this kind of data and saturate the existing benchmarks such that a quantitative comparison of models becomes difficult.

How to scale such methods to visually complex real-world data remains an open problem. When analyzing the current state-of-the-art methods and datasets, it becomes clear that there is a strong reliance on simple appearance (e.g., single color, simple shape). For example, [Greff et al. 2019] identify a tendency of their model to segment by color, and it fails when applied to natural images. In fact, the majority of methods learn semantic objects using similar compositional principles, which exploit statistical advantages in aligning simple scene elements with internal representations. Natural images and the objects therein, however, do not possess strong, consistent colors. Instead, they feature confounding textures, often a mixture of repeating and irregular patterns, which might violate such assumptions.

This work introduces a dataset and benchmark as the next step towards eventually tackling real-world scenarios. We propose ClevrTex, a synthetic dataset that consists of *textured* foreground objects and background, unlike existing benchmarks. Interestingly, we find that simply moving from uniformly colored to textured objects poses extreme challenges for current models, and no existing method achieves satisfactory performance. For this reason, we also introduce several variants of our dataset to gradually scale the visual complexity of the scenes and investigate where current algorithms struggle. To probe the generalization capability of models to out-of-distribution scenes, we create additional test sets that contain unseen shapes and materials and camouflaged objects. Together with ClevrTex and its variants, we are releasing the code to generate the dataset from scratch. Finally, we find that existing work does not rely on a consistent set of metrics and benchmarks. In an extensive set of experiments, we benchmark the majority[1] of current work on both CLEVR and our newly introduced ClevrTex.

[1]wherever code was available or could be obtained from the authors



GQN [S. A. Eslami et al. 2018] | ObjectRoom [C. P. Burgess et al. 2019] | ShapeStacks [Groth et al. 2018] | CLEVR [Johnson et al. 2017] | ClevrTex

Figure 2.1: Qualitative comparison of our new ClevrTex dataset with previous unsupervised multi-object learning datasets featuring 3D objects. See 2.1 for quantitative comparison.

Table 2.1: Comparison of the proposed ClevrTex dataset with previous unsupervised multi-object learning datasets featuring 3D objects.

Dataset	#Images	#Objects	#Shapes	#Obj. Colors	#Obj. Materials	#Backgrounds	Annotations
GQN [S. A. Eslami et al. 2018]	12M	1–3	7	—	1	15	Camera parameters
ObjectRoom [C. P. Burgess et al. 2019]	1M	1–6	4	10	1	100	Semantic, factor of variation
ShapeStacks [Groth et al. 2018]	310k	2–6	4	5	1	25	Semantic, stability, stability type
CLEVR [Johnson et al. 2017]	100k	3–10	3	8	2	1	Semantic, factors of variation
ClevrTex (Ours)	50k+10k	3–10	4+4	—	60+25	60+25	Semantic, depth, normal, shadow, factors of variation

2.2 Related Work

Object recognition benchmarks such as PascalVOC [Mark Everingham et al. 2010] or MS COCO [T.-Y. Lin et al. 2014] have been fundamental to object detection research. However, the current unsupervised multi-object segmentation models are yet unable to handle diverse real-world images featured in such datasets and have relied on visually trivial 2D and 3D data. Here, we review datasets and benchmarks used in *unsupervised* multi-object segmentation methods and point out the differences to ClevrTex.

2D Datasets Earlier unsupervised multi-object learning approaches were applied to transformed versions of existing 2D datasets, often originally crafted for disentanglement research, such as Shapes [Reichert et al. 2011], variants of MNIST [LeCun et al. 1998]: TexturedMNIST [Greff et al. 2016] and MultiMNIST [Sabour et al. 2017], as well as the multi-object version of dSprites [Matthey et al. 2017], i.e., Multi-dSprites [C. P. Burgess et al. 2019]. Others borrow data from the reinforcement learning community, such as the ATARI game environment [Bellemare et al. 2013] or Tetrominoes [Bozkurt et al. 2019]. However, 2D datasets, whilst valuable for development, do not contain the visual cues and details (e.g. shadows and perspective) needed to learn object segmentation that generalizes to real images.

3D Datasets Simple 3D Phong-shaded datasets (fig. 2.1) have been crafted for use in the unsupervised multi-object setting. The object-room dataset [C. P. Burgess et al. 2019], a multi-object extension of 3D shapes [C. Burgess and H. Kim 2018], features colored shapes arranged in a room with colored walls. ShapeStacks [Groth et al. 2018] features stacked, solid-colored primitives on a simple background with a pattern. CLEVR [Johnson et al. 2017], which is most closely related to our work, was introduced as a visual question-answering dataset but has become a popular benchmark in unsupervised scene decomposition as well. It features a set of 3-10 primitive shapes arranged on a gray photo backdrop; objects can have either a rubbery or metallic appearance and one of 8 color tints. CLEVR6 [Greff et al. 2019] is a filtered version of the CLEVR dataset that includes only up to 6 objects per image. It is often used for training in multi-object representation learning, with the remainder of CLEVR used to test generalization to more crowded scenes [Locatello et al. 2020; Emami et al. 2021].

Additional variants of CLEVR have also been generated for other tasks, such as ARROW [Jiang and Ahn 2020] for exploring scene composition accuracy, and a recursive version in [F. Deng et al. 2021] for learning part-whole relationships. Multi-view variations [Kosiorek et al. 2021; Stelzner et al. 2021] are used for 3D representation learning, and further include new object geometry, such as toys [N. Li et al. 2020a] and chairs [Yu et al. 2021]. However, these datasets feature simple scenes of low visual complexity, with contrasting solid colors present on objects. ClevrTex instead contains difficult objects with various materials that include repeating patterns and small details and often blend in rather than stand out from the background.

The main differences in data statistics between ClevrTex and commonly used multi-object learning datasets are also summarised in table 2.1.

Unsupervised Multi-Object Segmentation in Natural Scenes Some attempts have also been made to scale to natural scenes. [S. M. A. Eslami et al. 2016] apply the AIR model modified with a 3D rendering engine to infer identities and positions of crockery items on a table, training on simulated data, and evaluating against real-world images. [Monnier et al. 2021] test their sprite-based method on foreground/background segmentation on the Weizmann Horse dataset [Borenstein

and Ullman 2004]. [Engelcke et al. 2021] apply Genesis-V2 to robotic manipulation datasets, Sketchy and APC [A. Zeng et al. 2017]. Sketchy [Cabi et al. 2019] features recordings of a robotic arm manipulating solid colored toys, towels, or other small objects on a test table, but it lacks segmentation masks. The APC [A. Zeng et al. 2017] dataset is used instead for evaluation but only contains a single foreground object. These attempts signal promise that unsupervised multi-object segmentation can eventually scale to diverse real-world images.

Visual Fidelity in Simulation Simulation has always been central to progress in machine/reinforcement learning. However, as usual, the gap between a simulated setting and the ability to generalize to real-world environments is of concern. Several new simulators aim to improve the visual fidelity using photo-mapped environments or artists' compositions [Savva et al. 2017; Kolve et al. 2017; F. Xia et al. 2018; Manolis Savva* et al. 2019]. Recently, TWD [Gan et al. 2021] introduced a rich physics engine and PBR rendering of environments with a library of objects. Similar to our work, the emphasis is partly on increasing visual fidelity while moving away from trivial settings and towards real-world applications. However, RL environments have not seen much use in the unsupervised vision domain due to the often specific nature of the data, egocentric perspective, and temporal dependency.

2.3 ClevrTex

We introduce ClevrTex, a simulated dataset designed to present the next challenge in unsupervised multi-object learning. It introduces confounding visual aspects such as texture, irregular shapes, and various materials while maintaining control over scene composition. ClevrTex is available under CC-BY license.

2.3.1 Dataset Creation

ClevrTex is a much more visually complex extension of CLEVR [Johnson et al. 2017] targeted at multi-object learning. It is procedurally generated using the API of Blender[2], a powerful open-source 3D suite.

[2] https://www.blender.org/

At the center of the CLEVRTEX generation process is a catalog of diverse photo-mapped materials[3] ranging from forest floor duff, rocks, brickwork, and tiles to fabrics, metallic weaves, and meshes — a full list of materials is shown in appendix B.3.5). To generate each image, we start with a scene containing only a photo backdrop, which will become the background. For viewpoint and lighting diversity, we apply random jitter to the position of the camera and three lights. We then fill the scene with 3 to 10 objects (number sampled uniformly), sampling each object from a set of shapes: a cube, a sphere, a cylinder, and a non-symmetric shape of anthropomorphized monkey head[4] for increased complexity in object silhouettes. Objects are added to the scene one by one by sampling position (continuous, $(x, y) \sim \text{Uniform}(-3, 3)$), scale (discrete, $s \in \{.9, .6, .4\}$), and rotation (continuous, $\theta \sim \text{Uniform}(0, 360)$). If a new object collides with already existing shapes in the scene, the object's transformation is resampled until no collision is found or a maximum number of trials is exceeded, at which the process restarts by removing all objects.

We then sample a material for each object and the background. Using adaptive subdivision, we create material-specific geometry by displacing vertices of the starting shapes. This creates reliefs for simpler materials or distorts shapes, extruding features or introducing holes. The materials use albedo, subsurface scattering, and reflectivity maps to generate detailed visuals. Using physically based rendering ensures appropriately detailed reflections, highlights, and lighting effects. In addition, we generate ground truth segmentation maps through the rendering process and automatically check that no object is fully occluded. In that case, the scene is resampled from scratch. Further figures depicting scene lighting, objects, their scales and deformations are available in appendix B.3.5.

Figure 2.2: CLEVRTEX and its variants.

[3]We use the computer graphics term "material" to refer to the collection of resources used to creates the likeness of appropriate real-world material on simulated surfaces. Materials are typically a composition of various modalities, such as normal, diffuse, specular, and displacement maps, as well as a computation graph and shaders. We use the term "texture" to refer to 2D images mapping color information onto 3D surfaces.

[4]A modified version of Suzzane – a prefab shape available in Blender.

The object shapes and placement mimics that of the CLEVR dataset [Johnson et al. 2017] for backward compatibility. We do not generate the question-answering part of the original CLEVR dataset but include full metadata. This means that this dataset could also be used for other CLEVR-based tasks such as question answering, although this is not our focus here. Similarly, in anticipation that our dataset might also find usages beyond its intended setting, we include depth, albedo, shadow, and normal maps alongside the images, segmentation maps, and metadata. We share the code to generate ClevrTex alongside the dataset.

2.3.2 Statistics

ClevrTex contains 50 000 images, of which we use 10% for testing, 10% for validation and the remaining 80% (40 000 images) for training. Each image contains between three and ten objects (uniformly sampled). There are four possible shapes, which have been modified to enable clean texture mapping. We use three distinct object scales to maintain identifiable size "names", as in CLEVR, and custom meshes to ensure that the scaling of the objects does not distort texture details. The object placement and rotation are sampled from a continuous range. Note that even though two shapes — cylinder and sphere — are rotationally symmetric, the materials applied to them are not. We use a catalog of 60 materials with non-commercial licenses to generate the whole dataset before splitting the data into training sets. The materials are manually adjusted to ensure visually pleasing results at different scales and the background.

2.3.3 Variants

We create the following modifications of ClevrTex, each with 20 000 images (see fig. 2.2), to enable a more detailed analysis and evaluation and probe methods for their shortcomings.

The first variant, PlainBG, is a dataset consisting of textured objects on a plain background, i.e., the background is always set to a simple material as in CLEVR. We also create the reverse version, VarBG (varied background), where the objects are assigned simple CLEVR-like materials and colors while the background receives a textured material at random from our material catalog. PlainBG and VarBG fall in-between CLEVR and ClevrTex in terms of

visual complexity. In PlainBG, intra-object appearance is more complex, but each object clearly stands out from the plain background. On the other hand, VarBG maintains uniformly colored objects but introduces background texture, effectively making the background more diverse than the foreground. PlainBG and VarBG can be used to analyze the importance of background vs. object reconstruction. Furthermore, we create GrassBG, which contains scenes with the same mossy grass material as the background, while foreground objects receive materials at random. This variant is thus comparable to ClevrTex in terms of visual complexity. However, consistency in the background allows for testing memorization vs. reconstruction effects.

In addition, we propose the following two test sets to serve as an extra check for the limitations of ClevrTex.

CAMO contains scenes with "camouflaged" objects. To simulate this, every scene is made of a single, randomly sampled material that is used on all objects *and* the background. CAMO is created to challenge the internal-vs-external consistency and the efficiency hypothesis that underpins compositional methods. The only visual cues here are lighting, shadows, and perspective. It should enable probing models to see if they rely on such context to identify objects. Although we release CAMO with training, validation and test splits, in our experiments it is only used as a testbed for models trained on ClevrTex.

Finally, we also provide a separate OOD (out-of-distribution) dataset to evaluate model generalization on novel scenes. This dataset is designed exclusively as a test set and thus only contains 10 000 images. OOD is generated the same way as ClevrTex, but exclusively uses 25 *new* (unseen) materials — i.e. different from the 60 already used in other variants — and four new shapes (cone, torus, icosahedron, and a teapot) that are not part of ClevrTex.

2.4 Models

In recent years, there has been a surge of methods that aim to decompose a scene into objects in an unsupervised manner and, at the same time, learn object-centric representations. Following [Z. Lin et al. 2020b], we categorize these methods as follows.

Pixel-Space Approaches (⊞) A common way to frame the problem of unsupervised scene decomposition into objects is to assign each pixel to one of a usually fixed number of scene components, inferring per-pixel membership maps [Greff et al. 2016; Greff et al. 2017; Greff et al. 2019; C. P. Burgess et al. 2019; Yanchao Yang et al. 2020; Emami et al. 2021]. While these methods are probabilistic in nature, they do not lend themselves to generating new images. For this reason, several generative methods have been proposed, where images can be sampled from the learned distributions [Engelcke et al. 2020; Engelcke et al. 2021]. Finally, [Locatello et al. 2020] introduce a discriminative approach using an iterative clustering-like slot attention mechanism.

Here, we benchmark MONet [C. P. Burgess et al. 2019] and IODINE [Greff et al. 2019] as examples of earlier approaches that handle 3D colored scenes. We also evaluate the improved efficient MORL (eMORL) [Emami et al. 2021], Genesis-v2 [Engelcke et al. 2021] as a generative model, and Slot Attention [Locatello et al. 2020] which is representative for discriminative models.

Glimpse-Based Methods (⊡) An alternative to predicting components for each pixel is to extract patches of the input—named *glimpses*—that contain objects. A dense segmentation can be derived in this reduced space. These glimpses are arranged on top of an explicit background to reconstruct the image. Glimpse-based methods [S. M. A. Eslami et al. 2016; Crawford and Pineau 2019; Z. Lin et al. 2020b; Jiang and Ahn 2020; F. Deng et al. 2021] tend to offer computational advantages due to smaller regions, however also require deciding, extracting and composing patches.

Table 2.2: Computational resources for different models. × indicates number of GPUs needed. Measured on NVIDIA P40 24GB GPUs, with original batch sizes and 128 × 128 input. Train. time refers to time required to train the models for the recommended number of iterations, measured in total GPU hours. Inf. time measures the mean inference time required for a single batch, shown $\pm\sigma$ over 7 passes.

Model	Train. Time (GPU h)	Inf. Time (ms$\pm\sigma$)	Peak GPU Mem (GB)
⊡ GNM [Jiang and Ahn 2020]	54	258 ±9	4
⊡ SPACE [Z. Lin et al. 2020b]	64	191 ±2	8
⊡ SPAIR* [Crawford and Pineau 2019]	77	213 ±2	11
◘ DTI [Monnier et al. 2021]	198	2530 ±5	11
◘ MN [Smirnov et al. 2021]	—	—	11
⊞ IODINE [Greff et al. 2019]	4 × 202	1360 ±2	4 × 23
⊞ SA [Locatello et al. 2020]	290	818 ±1	17
⊞ MONet [C. P. Burgess et al. 2019]	3 × 106	544 ±1	3 × 17
⊞ eMORL [Emami et al. 2021]	4 × 158	217 ±1	4 × 17
⊞ GenV2 [Engelcke et al. 2021]	194	452 ±1	15

From the glimpse-based methods, we benchmark SPAIR [Crawford and Pineau

2019], which models glimpses auto-regressively, using a truncated geometric prior. Since it cannot handle non-black backgrounds, we modify the model to include a VAE for background prediction (SPAIR*). We also evaluate SPACE [Z. Lin et al. 2020b] due to its use of the pixel-space approach for processing the background, and GNM [Jiang and Ahn 2020], which uses scene-level priors.

Sprite-Based Methods (🃏) Recently, several methods [Smirnov et al. 2021; Monnier et al. 2021] propose to decompose images into a learned dictionary of RGBA sprites instead of learning a generative model. From the alpha masks of each sprite, the scene segmentation can be recovered. We benchmark MarioNette [Smirnov et al. 2021] and DTISprites [Monnier et al. 2021] to investigate the differences of two sprite-based (🃏) approaches.

The aforementioned models have highly varying computational requirements. We offer a side-by-side comparison in table 2.2, where computational advantages to glimpse-based methods can be immediately seen, with methods such as GNM and SPACE taking a fraction of time and memory required by even single-GPU pixel-space methods. All implementation details, hyper-parameters, and model changes are reported in appendix B.3.3.

2.5 Experiments

Datasets We benchmark a wide spectrum of methods using CLEVRTEX and its variants. To test generalization, we evaluate models trained on CLEVRTEX using OOD and CAMO. In addition to our CLEVRTEX and its variants, we conduct experiments on CLEVR to provide a complete side-by-side comparison of methods and the new challenges in CLEVRTEX. All implementation details and preprocessing are reported in appendix B.3.1.

Metrics The majority of previous work has used the adjusted Rand index on foreground pixels (ARI-FG) only as an evaluation metric. We share concerns with [Monnier et al. 2021; Engelcke et al. 2020] that this metric does not reflect how well objects are localized by the model and whether they are considered part of the background. Thus, we report mean intersection over union (mIoU) instead, as it considers the background. Further discussion and a side-by-side comparison of

ARI-FG and mIoU can be found in appendix B.3.2. Furthermore, we judge the quality of the reconstruction output of the models using the mean squared error (MSE). For the models trained on CLEVR and CLEVRTEX, we report results on three random seeds, including their standard deviation.

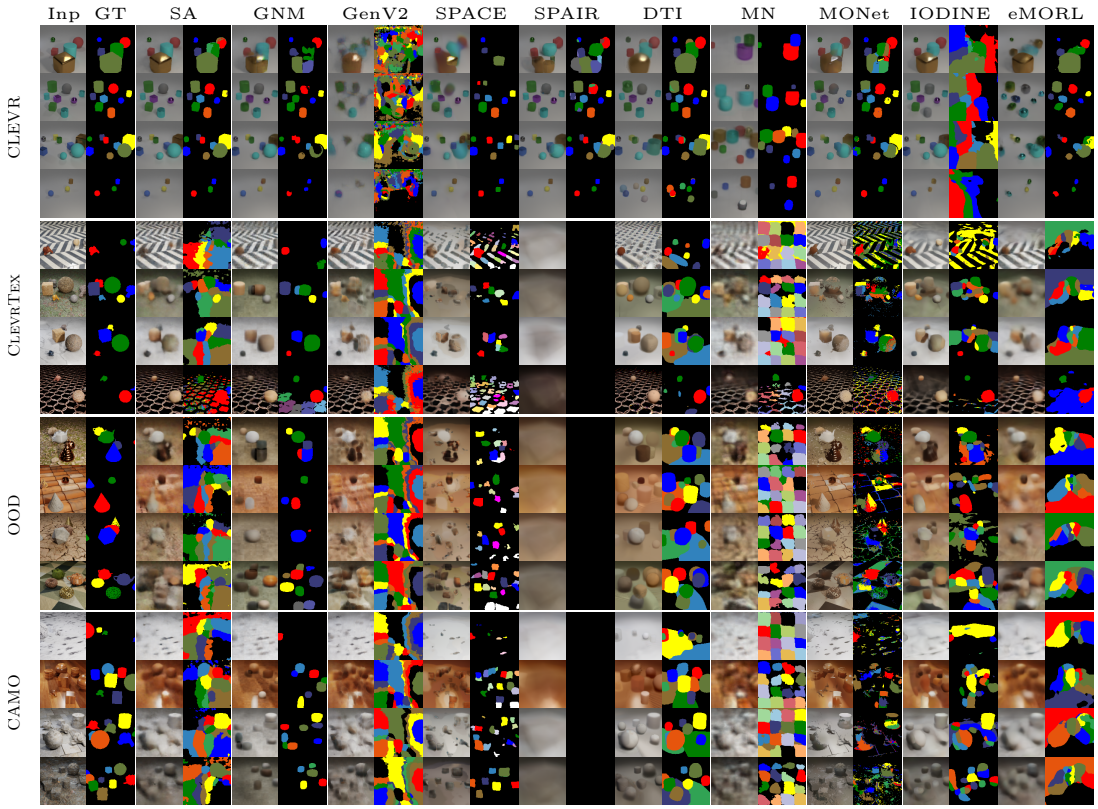


Figure 2.3: Comparison of various models' reconstruction and segmentation outputs on CLEVR, CLEVRTEX and our test sets. Best viewed digitally. More results in the Appendix, fig. B.1.

2.5.1 Benchmark

The results for the benchmark are detailed in table 2.3 and in fig. 2.3. Next, we discuss our findings regarding the ability of models to separate foreground and background, to handle textured scenes, as well as their training stability and generalizability to new scenes.

Background Segmentation Pixel-space methods (⊞) show impressive performance on CLEVR compared against glimpse-based approaches (⊡) on the foreground (see fig. 2.3). However, if we consider the ability to segment the background (mIoU in table 2.3), their performance advantage disappears, with SPAIR* performing the best. We attribute this to the tendency of pixel-space models

Table 2.3: Benchmark results on CLEVR and CLEVRTEX and the generalization test sets CAMO, and OOD. Results shown $\pm\sigma$ calculated over 3 runs. † updated eMORL: after CLEVRTEX was released, the authors of [Emami et al. 2021] have updated their codebase to include CLEVRTEX training and evaluation and shared their trained models with improved performance (single seed on CLEVR).

Model	CLEVR		CLEVRTEX		OOD		CAMO	
	↑mIoU (%)	↓MSE	↑mIoU (%)	↓MSE	↑mIoU (%)	↓MSE	↑mIoU (%)	↓MSE
⊡ SPAIR* [Crawford and Pineau 2019]	65.95± 4.02	55± 10	0.0 ± 0.0	1101± 2	0.0 ± 0.0	1166± 5	0.0 ± 0.0	668± 3
⊡ SPACE [Z. Lin et al. 2020b]	26.31±12.93	63± 3	9.14± 3.46	298± 80	6.87± 3.32	387± 66	8.67± 3.50	251± 61
⊡ GNM [Jiang and Ahn 2020]	59.92± 3.72	43± 3	42.25± 0.18	383± 2	40.84± 0.30	626± 5	17.56± 0.74	353± 1
♣ MN [Smirnov et al. 2021]	56.81± 0.40	75± 1	10.46± 0.10	335± 1	12.13± 0.19	409± 3	8.79± 0.15	265± 1
♣ DTI [Monnier et al. 2021]	48.74± 2.17	77± 12	33.79± 1.30	438± 22	32.55± 1.08	590± 4	27.54± 1.55	377± 17
⊞ GenV2 [Engelcke et al. 2021]	9.48± 0.55	158± 2	7.93± 1.53	315±106	8.74± 1.64	539±147	7.49± 1.67	278± 75
⊞ eMORL [Emami et al. 2021]	50.19±22.56	33± 8	12.58± 2.39	318± 43	13.17± 2.58	471± 51	11.56± 2.09	269± 31
⊞ eMORL† [Emami et al. 2021]	21.98	26	30.17± 2.60	347± 20	25.03± 1.99	546± 4	19.13± 4.88	315± 21
⊞ MONet [C. P. Burgess et al. 2019]	30.66±14.87	58± 12	19.78± 1.02	146± 7	19.30± 0.37	231± 7	10.52± 0.38	112± 7
⊞ SA [Locatello et al. 2020]	36.61±24.83	23± 3	22.58± 2.07	254± 8	20.98± 1.59	487± 16	19.83± 1.41	215± 7
⊞ IODINE [Greff et al. 2019]	45.14±17.85	44± 9	29.16± 0.75	340± 3	26.28± 0.85	504± 3	17.52± 0.75	315± 3

to assign parts of the background to nearby objects. In glimpse-based methods, however, the formation of glimpses forces the objects to be spatially compact, which offers an advantage when separating the objects from the background.

Textured Scenes When training on CLEVRTEX, all models struggle. The foreground segmentation performance reduces, indicating that models fail to assign whole objects to a single component, likely due to the tendency to overfit consistent color regions. The overall segmentation performance is worse as well. MSE is much higher than on CLEVR, with models producing blurry or flat reconstructions, failing to capture much of the rich variation in the input data. SPAIR*, which showed the best overall performance on CLEVR, fails to recognize any objects and instead simply predicts the background. We conjecture that SPAIR's autoregressive handling of objects paired with the use of spatial transformers might make the learning signal too noisy.

Sprite-based models (♣) also perform worse, as the greater variation in appearances is not sufficiently captured by their limited dictionary. While the dictionary size can be increased, the lack of an internal compression mechanism to represent varied appearances will always be a limiting factor in natural world settings. Interestingly, when unable to capture individual objects, MN learns to tile the image with possible color blobs, representing low-frequency information in the image instead. In our tests, similar tiling behavior tends to occur also in glimpse-based models whenever they cannot learn to reconstruct the foreground (see the Appendix, fig. B.2, for examples in other models). Since DTI includes a set of internal transformations, it

performs comparatively better on ClevrTex.

GNM, a generative glimpse-based approach, has overall the best performance on ClevrTex, which we attribute to spatial-locality constraints imposed through the glimpse-based formulation and limited background reconstruction ability due to a simpler background model; i.e. comparing to other methods less capacity is spent on the background. Interestingly, GNM shows one of the largest reconstruction errors, despite being the best at scene segmentation, suggesting that ignoring confounding aspects of the scene rather than representing them might aid in the overall task.

Out of the our benchmarked pixel-space methods (⊞), IODINE performs the best in terms of the overall segmentation performance. Our qualitative investigation shows that pixel-space methods that can segment textured scenes largely capture consistent color regions, which occasionally align with objects on scenes with simpler materials. Large patterns in the background or changes in object appearance, often due to lighting result in oversegmentation.

Stability Due to inherent stochasticity in initialization and optimization, one can expect a degree of variation between different model training runs. Many benchmarked models in this study also rely on internal randomness, primarily due to the sampling procedures involved. This influences the learning signal and the configuration the models can learn. Pixel-based approaches and SPACE (which has pixel-space model for background) show higher variance in the performance metrics. Similar to [Locatello et al. 2020; Emami et al. 2021; Z. Lin et al. 2020b], we observe that these methods occasionally fail to use separate components, which causes high fluctuation between different seeds. Glimpse-based methods are more stable with respect to seeds but tend to exhibit higher sensitivity to hyperparameter settings.

Generalisation In addition to benchmarking existing approaches in their ability to learn and handle textured scenes, we are also interested in the degree to which different approaches might rely on specific factors of ClevrTex. To this end, we evaluate the models trained on the ClevrTex on two additional test sets: CAMO to see whether models rely on the difference of object appearances present in a scene, and OOD to see whether a degree of memorization (e.g. of shapes and materials) plays a role in recognition and whether the methods could generalize to

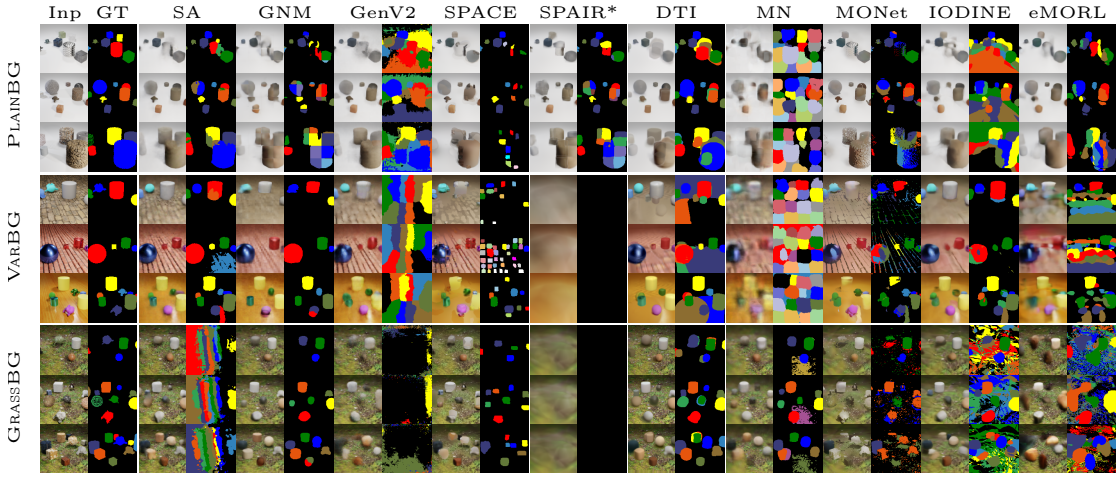


Figure 2.4: Comparison of various models' reconstruction and segmentation outputs on PlainBG, VarBG and GrassBG variants. Best viewed digitally.

unseen patterns.

Interestingly, some of the better performing approaches on ClevrTex maintain much of their segmentation ability on out-of-distribution (OOD) data. GNM, for example, attempts to reconstruct the input using memorized training data materials and shapes, which leads to reduced but still comparable object segmentation. Other sprite- (🂡) and glimpse-based (⊡) methods either do not perform well or show similar reliance on the appearances from the training distribution. Pixel-space models (⊞) show a better ability to reconstruct the input but also tend to reconstruct based on consistent color regions rather than objects, a tendency only exacerbated by the out-of-distribution setting.

When considering the challenging CAMO setting, none of the approaches perform satisfactory segmentation. Methods that somewhat work on ClevrTex tend to use different components to represent lighter and darker parts of the scene, highlighting the tendency of all current models to overfit the scene appearance.

2.5.2 Variants

As discussed above, many of the models that perform well on CLEVR, either do not work on ClevrTex or lose much of their performance. To further probe which aspects of the scene composition are challenging, we use the variants of ClevrTex.

Table 2.4: Model results on PlainBG,VarBG, and GrassBG variants.

Model	PlainBG		VarBG		GrassBG	
	↑mIoU (%)	↓MSE	↑mIoU (%)	↓MSE	↑mIoU (%)	↓MSE
⊡ SPAIR* [Crawford and Pineau 2019]	39.32	134	0.00	1246	0.00	728
⊡ SPACE [Z. Lin et al. 2020b]	31.96	120	16.10	311	33.85	196
⊡ GNM [Jiang and Ahn 2020]	26.49	96	49.78	438	53.15	254
♣ MN [Smirnov et al. 2021]	10.16	167	11.51	441	34.80	266
♠ DTI [Monnier et al. 2021]	36.03	210	38.82	498	37.65	215
⊞ GenV2 [Engelcke et al. 2021]	24.39	98	14.40	298	2.88	306
⊞ eMORL [Emami et al. 2021]	29.39	96	22.92	385	19.38	199
⊞ MONet [C. P. Burgess et al. 2019]	38.72	83	23.73	212	21.29	165
⊞ SA [Locatello et al. 2020]	39.32	134	62.57	257	12.88	116
⊞ IODINE [Greff et al. 2019]	23.83	128	39.86	364	25.76	225

Textured Objects When applied to PlainBG, where materials are only seen on objects, and the background is gray, all of the methods still perform worse than on CLEVR, with a significant drop in segmentation performance, especially prevalent in pixel-space approaches (⊞). Since all methods have been designed with simpler datasets and uniformly colored objects, the more realistic nature of the materials in ClevrTex poses a difficult challenge. Glimpse-based models (⊡) also show reduced segmentation quality over CLEVR. MN (sprite-based) struggles as the increased diversity in foreground objects overwhelms the spite dictionary. Finally, the models' inability to capture the fine-grained details of the more complex object appearance causes the increase in reconstruction error.

Textured Background VarBG contains simple mono-colored objects arranged on top of a diverse set of textured backgrounds. Certain models, like SPAIR*, SPACE, and GenV2, struggle to handle diverse backgrounds. Other methods, however, seem to benefit from simpler objects, showing improvements in segmentation performance over both PlainBG and ClevrTex scenarios, indicating that these models rely on simpler, more consistent objects.

Consistent Background GrassBG has the same complex forest grass background in all scenes. The background is richer and more complex than in PlainBG. As glimpse-space methods (⊡) tend to model the background explicitly, we observe that contrasting consistent background aids these models greatly. Pixel-space methods (⊞) also perform slightly better in this setting than on ClevrTex where the background varies. However, the effect is not as pronounced as for glimpse-based (⊡) approaches, with the overall performance roughly matching what was observed

on ClevrTex.

2.6 Conclusions

Unsupervised object learning and scene segmentation is a challenging task. Interestingly, given the existing metrics and commonly used datasets (e.g., CLEVR), current approaches show impressive performance, yet we have shown that they are easily challenged when visual complexity increases. To this end, we present ClevrTex, a new benchmark that aims to increase visual scene complexity, which contains richer textures, materials, and shapes, to encourage progress towards methods applicable to real images in the wild.

In our experiments, GNM [Jiang and Ahn 2020] and IODINE [Greff et al. 2019] perform the best out of glimpse-based and pixel-space models, respectively, with GNM showing the best segmentation performance overall. However, almost all methods struggle to handle multiple textured scenes, resulting in a significant performance gap with respect to the closest current benchmark, CLEVR. Our findings suggest that pixel-space methods tend to be more prone to overfitting consistent color regions and smooth gradients. On the other hand, sprite- and glimpse-based approaches tend to memorize small repeated patterns, which offers an advantage on ClevrTex. Further testing, however, shows that these models reconstruct smooth backgrounds and recognize sharp changes as objects. As such, even the approaches that show some ability to handle textured environments focus largely on scene appearance, failing to learn and exploit global context clues that might align with semantic objects.

We believe that textures pose a challenge to current pixel-space and glimpse-based methods as they are built to exploit simple visual elements and uniform appearance that is present in previous datasets, partly due to the reconstruction objectives. We find evidence for this in our experiments with the dataset variants: consistency within *objects*, as seen in our VarBG variant, and consistency in *backgrounds* (PlainBG and GrassBG) helps to learn better models than the full ClevrTex where there is no simple intra- and inter-appearance consistency. Only on simpler scenes (fig. 2.3) the best performing methods succeed at segmenting some objects.

Thus, ClevrTex offers new challenges for unsupervised multi-object segmentation,

especially for evaluating generalization. Furthermore, the three variants and two additional test sets can serve as a diagnostic tool for developing new methods, and the extensive evaluation acts as a standardized benchmark for current and future methods.

Limitations The proposed dataset contains a limited number of primitive shapes and a catalog of 60 materials. Although future models might exploit the non-exhaustive nature of object appearance, e.g., memorizing object reconstructions than learning generalizable scene decompositions, we have shown that current methods are, in fact, faced with a significant challenge, even at a slight increase of data complexity (e.g., on PlainBG). To further address this limitation, we have created the OOD dataset, which should serve as an additional test for the generalization ability of models outside the training distribution. Overall, ClevrTex is still a synthetic dataset and does not fully close the gap to real-world data. However, until methods can solve ClevrTex, generalization to real images is likely out of reach.

Broader Impact The work presented here critically evaluates current approaches for unsupervised multi-object segmentation. The introduced datasets are fully simulated renderings of 3D primitives and do not contain any people or personal information. Our benchmark aims to establish and standardize evaluation practices, provide new challenges for current algorithms, and help future research compare with prior work. While ClevrTex is highly important for current research, its impact outside of the research community is low as current methods can not yet properly deal with real images.

Acknowledgments L. K. is funded by EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems EP/S024050/1. I. L. is supported by the European Research Council (ERC) grant IDIU-638009 and EPSRC VisualAI EP/T028572/1. C. R. is supported by Innovate UK (project 71653) on behalf of UK Research and Innovation (UKRI) and by the ERC IDIU-638009. We thank [Johnson et al. 2017] for their open-source implementation of CLEVR. We would also like to thank Martin Engelcke for helpful suggestions on applying Genesis-V2 to ClevrTex, Patrick Emami for assistance adapting eMORL to ClevrTex and

Dmitriy Smirnov for sharing their implementation of MarioNette.

Statement of authorship can be found in appendix A. Supplementary material is available in appendix B.

Chapter 3

Guess What Moves: Unsupervised Video and Image Segmentation by Anticipating Motion

In the previous chapter, we explored learning from the appearance of the scene but found that existing methods were limited in their ability to handle visually complex scenes. In this chapter, we explore an alternative source of information based on the principle of common fate (P3). We investigate how observation of motion expressed as optical flow can be used to learn segmentation of objects. Rather than relying on motion alone, we combine it with appearance information. We propose to learn an appearance-based binary segmentation model for real-world scenes by using optical flow as a source of supervision. To enable this, we propose a loss function that ties together the segmenation of the scene with the optical flow. We show that this approach can be used to learn segmentation of objects in real-world scenes for both video and image binary segmenation settings.

This chapter presents a paper that has been published in *British Machine Vision Conference (BMVC), 2022*. Additionally, it has been awarded a Spotlight presentation.

Guess What Moves: Unsupervised Video and Image Segmentation by Anticipating Motion

Subhabrata Choudhury*, Laurynas Karazija*,
Iro Laina, Andrea Vedaldi, Christian Rupprecht
Visual Geometry Group
University of Oxford
Oxford, UK
{subha,laurynas,iro,vedaldi,chrisr}@robots.ox.ac.uk

Abstract

Motion, measured via optical flow, provides a powerful cue to discover and learn objects in images and videos. However, compared to using appearance, it has some blind spots, such as the fact that objects become invisible if they do not move. In this work, we propose an approach that combines the strengths of motion-based and appearance-based segmentation. We propose to supervise an image segmentation network with the pretext task of predicting regions that are likely to contain simple motion patterns, and thus likely to correspond to objects. As the model only uses a single image as input, we can apply it in two settings: unsupervised video segmentation, and unsupervised image segmentation. We achieve state-of-the-art results for videos, and demonstrate the viability of our approach on still images containing novel objects. Additionally we experiment with different motion models and optical flow backbones and find the method to be robust to these change. Project page and code available at https://www.robots.ox.ac.uk/~vgg/research/gwm.

*Authors contributed equally.

3.1 Introduction

The motion of objects in a video can be detected by methods such as optical flow and used to discover and segment them. A key benefit is that optical flow is object-agnostic: because it relies on low-level visual properties, it can extract a signal even before the objects are discovered, and can thus be used to establish an understanding of objectness.

The potential of motion as a cue is epitomized in *video segmentation* problems, where the input is a generic video sequence and the task is to extract the main object(s) in the video. In fact, some methods [Charig Yang et al. 2021; Meunier et al. 2022] adopt a motion-*only* approach to video object segmentation, arguing that motion patterns are much easier to model and interpret than appearance. However, this approach ignores appearance cues and is 'blind' to stationary objects.

Instead, we propose to use motion as *supervision* to discover objects in videos *and* still images without the need for manual annotations. We observe that different objects tend to generate distinctive optical flow patterns which can be well approximated by small parametric models, such as affine or quadratic. We use this fact to train a segmentation network that, given a *single RGB frame* as input, predicts *which image regions* are likely to contain such patterns. The idea is that these regions would then separate the objects from the background.

This approach has several useful properties. First, while motion is used for supervising the network, the latter implicitly learns the appearance of the objects, regularizing the segmentation. Second, because the network works with a single image as input, it does not observe the motion directly. The model must anticipate what *could* move, extracting objects even if they are not in motion. Third, the network avoids predicting the objects' motion directly, which is a highly-ambiguous task given a single image as input; instead, it predicts only the support regions of the motion patterns, and the training loss measures the compatibility of such regions with the observed motion according to the assumed motion model.

While we are not the first to consider motion as a cue for decomposing an image into objects, our particular way of modeling motion is simple and versatile, and allows two application modes of our approach. First, we consider *internal learning for unsupervised motion segmentation* [Ulyanov et al. 2020]. Given one or more videos

as input (without labels), we optimize a network, as described above, to output a segmentation of the videos, effectively 'observing' motion via backpropagation. Our approach achieves state-of-the-art performance on standard benchmarks for unsupervised motion segmentation [Yanchao Yang et al. 2019; Charig Yang et al. 2021].

The second mode is *transductive learning for unsupervised image segmentation*, which is intended to assess the generalization capabilities of our model as an image segmenter. In this case, the network is first trained on a number of training videos and then evaluated on a disjoint set of images. Since only appearance information is available at test time, the problem solved is not motion segmentation, but image segmentation. In this scenario, our model segments novel objects not observed during training, demonstrating the viability of our approach.

3.2 Related Work

Our work aims to combine motion and appearance cues for unsupervised object discovery, in that motion can be used as a cue to learn a general object segmenter for both videos and images. As such, there exist several related areas in literature, which we review next.

Unsupervised Video Object Segmentation. The aim of video object segmentation (VOS) is to densely label objects present in a video. Current VOS benchmarks [Perazzi et al. 2016; F. Li et al. 2013; Ochs et al. 2014] usually define the problem as foreground-background separation, where the foreground comprises the most salient objects. Efforts to reduce the amount of supervision follow two main directions, semi-supervised and unsupervised VOS. Semi-supervised methods require manual annotations for the object(s) of interest in an initial frame during inference; the goal is to re-localize these objects across the video [Caelles et al. 2017]. Unsupervised VOS aims to discover object(s) of interest without the initial targets [Faktor and Irani 2014; Papazoglou and Ferrari 2013; Tokmakov et al. 2019; Jain et al. 2017; S. Li et al. 2018; X. Lu et al. 2019]. However, most unsupervised VOS methods use, in fact, some form of supervised pre-training on external data.

Motion Segmentation. In videos, the background is usually relatively static whereas objects in the scene have independent motion, thus providing a strong 'objectness' signal. Thus, many works approach unsupervised video object segmentation as a motion segmentation problem. Several earlier methods address this problem by grouping point trajectories [Brox and Malik 2010b; Sundaram et al. 2010; Ochs and Brox 2012; Keuper et al. 2015; Keuper et al. 2020; Ochs and Brox 2011], motion boundaries [Papazoglou and Ferrari 2013], voting [Faktor and Irani 2014] and layered models [Chang and III 2013; Jojic and Frey 2001]. More recently, [Xie et al. 2022; Lamdouar et al. 2021] train motion models on generated scenes with synthetic 2D objects and generalize to real videos. CIS [Yanchao Yang et al. 2019] proposes an adversarial framework, where an inpainter is tasked with predicting the optical flow of a segment based on context, while the generator aims to create segments with zero mutual information such that the context becomes uninformative. DyStaB [Yanchao Yang et al. 2021] extends CIS using the segmentation output of a dynamic model to bootstrap a static one. In contrast to our method, this yields two separate models to choose from based on the application (i.e., video or static image segmentation). Instead, AMD [R. Liu et al. 2021] employs a single model with separate appearance and motion 'pathways' and performs unsupervised test-time adaptation for video segmentation. Finally, MG [Charig Yang et al. 2021] abandons the appearance pathway altogether, directly segmenting optical flow inputs with a Slot Attention-like architecture [Locatello et al. 2020].

Closer to our approach, another line of work uses various motion models to group image regions. Early methods [Jepson and Black 1993; Torr 1998] consider mixture models of flow to account for the fact that a region may contain multiple motion patterns. Another line of work [Bideau and Learned-Miller 2016; Bideau et al. 2018] segments object translation directions from motion angle field obtained by correcting for estimated rotation of the camera. [Mahendran et al. 2018] employ an affine flow model, using the entropy of flow magnitude histograms for loss to deal with noisy flow in real world. [Meunier et al. 2022] consider affine and quadratic motion models, however their method uses flows as input which makes it suitable only for videos during inference.

Unsupervised Image Segmentation. While we use motion as a learning signal, our method yields a general-purpose image segmentation network, separating an image into foreground and background, without using ground truth masks for supervision. Early work in unsupervised image segmentation makes use of handcrafted priors, e.g. color contrast [M.-M. Cheng et al. 2015; Y. Wei et al. 2012], while some recent methods also combine handcrafted heuristics to generate pseudo-masks and use them to train using deep networks [J. Zhang et al. 2018; Y. Zeng et al. 2019; Nguyen et al. 2019]. Others address this problem via mutual information maximization between different views of the input [Ji et al. 2019a; Ouali et al. 2020]. A recently emerging line of work [Bielski and Favaro 2019; M. Chen et al. 2019; Kanezaki 2018; Benny and Wolf 2020; Voynov et al. 2021; Melas-Kyriazi et al. 2022b] explores generative models to obtain segmentation masks. Many of them [Bielski and Favaro 2019; M. Chen et al. 2019; Kanezaki 2018; Benny and Wolf 2020] are based on the idea of generating foreground and background as separate layers and combine them to obtain a real image. Others [Voynov et al. 2021; Melas-Kyriazi et al. 2022b] analyze large-scale unsupervised GANs (e.g. BigGAN [Brock et al. 2019]) and find implicit foreground-background structure in them to generate a synthetic annotated training dataset. Alternative line of work explores feature maps of self-supervised Vision Transformers, such as DINO [Caron et al. 2021]. For example, STEGO [Hamilton et al. 2022] supports segmenting multiple *classes* in an image, performing *semantic* segmentation, by distilling features and class centroids from DINO. In [Melas-Kyriazi et al. 2022a] and TokenCut [Y. Wang et al. 2022], authors model image patches with an affinity graph based on DINO feature alignment and perform further analysis on this graph to extract masks. [Shin et al. 2022a] cluster features of a variety of self-supervised backbones to produce candidate masks, using them to train a segmenter. Instead, our model is trained on video data using optical flow as a supervisory signal. However, since it only requires a single image as input at test time, we show that our method is applicable to this task, providing an alternative approach to unsupervised object segmentation.

Unsupervised Object Discovery. While the above methods often aim to segment the most salient object(s) in an image, unsupervised multi-object segmentation explores the problem of decomposing a scene into parts, which typically include each individual foreground object and the background. The usual approach

is to learn structured object-centric representations, i.e. to model the scene with latent variables (slots) operating on a common representation [Greff et al. 2019; Locatello et al. 2020; Emami et al. 2021; Z. Lin et al. 2020b; Crawford and Pineau 2019; C. P. Burgess et al. 2019; Engelcke et al. 2020; Engelcke et al. 2021; Jiang and Ahn 2020; G. Singh et al. 2022a]. While these methods are image-based, extensions to video also exist [Kosiorek et al. 2018; Jiang et al. 2020; Crawford and Pineau 2020; Kabra et al. 2021; Zablotskaia et al. 2020; Besbinar and Frossard 2021; Min et al. 2021; Bear et al. 2020; Kipf et al. 2022; G. Singh et al. 2022b; Monnier et al. 2021]. These methods often operate in an auto-encoding fashion with inductive bias to separate objects derived from a reconstruction bottleneck [C. P. Burgess et al. 2019], that is often dependent on the architecture and the latent variable model. We similarly impose a reconstruction bottleneck on the flow but use a simple model grounded in projective geometry, with a known closed-form solution. It is also important to note that unsupervised *multi*-object segmentation appears to be significantly more challenging, with current methods exploiting the simplicity of synthetic scenes [Johnson et al. 2017; Girdhar and Ramanan 2020], while struggling on more realistic data [Karazija et al. 2021]. Recently, [Bao et al. 2022] explore an extension of Slot Attention [Locatello et al. 2020], guided by an external supervised motion segmentation algorithm, to real-world data. However, due to the difficulty of the problem, they operate in a constrained domain (autonomous driving) and consider only a limited number of object categories. We instead focus on wide variety of categories and settings encountered in common video segmentation datasets and consider both motion and appearance jointly.

3.3 Method

In this paper, we present a method that uses motion anticipation to discover and segment objects in images without the need for human annotations (overview in fig. 6.1). We use optical flow from video sequences as supervision for this problem. However, rather than predicting the flow directly, we task a general image segmentation network to predict image regions where motion may be explained by a simple coherent model. Such regions should align with optical flow patterns produced by objects that *could* move (but do not have to).

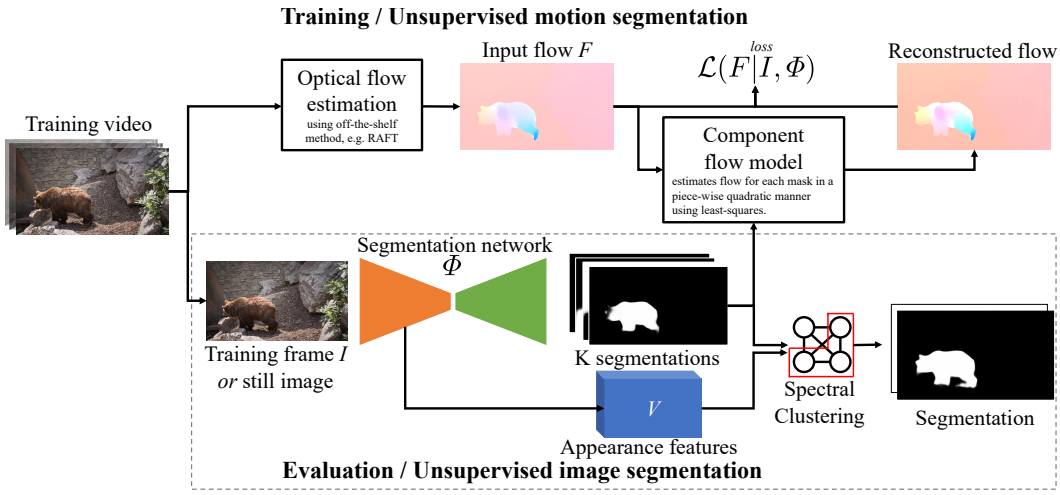


Figure 3.1: **Model Diagram.** We train a segmentation network to partition an image into K components without manual annotations. Our model is trained using individual frames from video as input and pre-computed optical flow as supervision. The predicted segments are used to approximate the input flow with piecewise quadratic flow models and the training loss is formulated as the error between the reconstructed and the input flow. Appearance features from the backbone are used to merge the predicted K segments into foreground and background components. Motion information is not required at test time and inference can be performed on still images. Optical flow is colorized for visualization only.

3.3.1 Segmentation by Motion Anticipation

Let $I \in \mathbb{R}^{3\times H\times W} = (\mathbb{R}^3)^\Omega$ be an RGB image defined on a lattice $\Omega = \{1,\dots,H\} \times \{1,\dots,W\}$. Assume that the image is a frame in a video sequence and let $F \in (\mathbb{R}^2)^\Omega$ be the corresponding optical flow (extracted from the video by means of an off-the-shelf optical flow network, such as RAFT [Teed and J. Deng 2020]). The goal is to decompose the image into K components (or regions), which is a classic segmentation problem. Hence, we learn a segmentation network $\Phi(I) \in ([0,1]^K)^\Omega$ that, given the image I as input, assigns each pixel u to one of K components in a soft manner, with probabilities:

$$P(m_u = k \mid I, \Phi) = [(\Phi(I))_k]_u, \quad u \in \Omega,\ k \in \{1,...,K\}. \tag{3.1}$$

$m_u = k$ in eq. (3.1) denotes the predicted mask corresponding to component k indexed by u. In particular, we seek to separate the foreground and background, for which one may choose $K = 2$, although as we show later (section 3.3.2), this need not be the case.

More specifically, we train Φ to partition pixels according to the Gestalt principle of common fate [Spelke 1990; Wagemans et al. 2012]. This is done by associating

each region $k \in \{1, ..., K\}$ to a model θ_k of the optical flow observed within it. That is, the optical flow corresponding to an input frame can be approximated by piece-wise parametric models, representing the motion or *flow pattern*, of each component independently. According to the common fate principle, pixels within the same region are expected to exhibit *coherent* motion.

A variety of motion models exist for describing the 2D flow of an object ([Mahendran et al. 2018; Meunier et al. 2022]). These are generally of the form $F_u \approx Au + b$, where parameters A, b can be recovered by solving a system of linear equations. One common choice is an affine model (where $u = [x, y]$ are pixel coordinates), which is sufficient if objects are smaller and further away from camera. The affine model, however, struggles if the depth of an object varies significantly resulting in more complex flow patterns. To factor out unknown depth information, each object can be modeled as a plane with a quadratic 8-parameter model [Adiv 1985]. Here, we allow for more complex geometry than planes, by using a simplified 12-parameter quadratic model $\theta_k = (A_k, b_k)$ with $A_k \in \mathbb{R}^{2\times5}$ and $b_k \in \mathbb{R}^2$ per region k. In this case, $u = [x,\ x^2,\ y,\ y^2,\ xy] \in \mathbb{R}^5$ includes quadratic and mixed terms of the pixel coordinates to model quadratic dependencies. The 12-parameter model also allows treating each flow direction independently. We assume that the model predicts the flow up to isotropic i.i.d. Gaussian noise, which results in a simple L^2 fitting loss:

$$-\log p(F_u \mid \theta_k) \propto \|F_u - A_k u - b_k\|^2. \tag{3.2}$$

Summing over all pixels, learning minimizes the energy function:

$$\mathcal{L}(F \mid \theta, I, \Phi) \propto \sum_{u\in\Omega} \sum_{k} \|F_u - A_k u - b_k\|^2 \cdot p(m_u = k \mid I, \Phi). \tag{3.3}$$

In the expression above, we *do not* know the flow parameters θ_k as the network only predicts the regions' extent. Instead, we *min-out* the parameters θ_k in the loss itself and compute

$$\mathcal{L}(F \mid I, \Phi) = \min_{\theta_{k\in\{1,...,K\}}} \mathcal{L}(F \mid \theta, I, \Phi). \tag{3.4}$$

The energy in eq. (3.2) is quadratic in θ_k, resulting in a weighted least squares problem that can be efficiently solved in closed form (see supplementary material).

Our model is learned from a large collection $\mathcal{T}$ of video frame-optical flow pairs (I, F), minimizing the empirical risk:

$$\Phi^* = \underset{\Phi}{\operatorname{argmin}} \frac{1}{|\mathcal{T}|} \sum_{(I,F)\in\mathcal{T}} \mathcal{L}(F \mid I, \Phi) \tag{3.5}$$

3.3.2 Over-segmentation

While the 12-parameter model is more powerful than an affine one, it is still not sufficient to model arbitrary flow patterns. In complex scenes that contain foreground and background clutter, we often observe motion parallax effects. Additionally, non-rigid objects and self-occlusions can result in complex flow patterns within the object that are not captured accurately by the quadratic model.

To account for such complexity, we propose to *over-segment* the input image into $K > 2$ regions. Over-segmentation enables the model to use additional regions to explain several moving objects and to approximate varyingly moving parts of a single non-rigid object as well as motion parallax. To achieve a binary segmentation output, one needs a criterion to merge a number of predicted regions down to foreground and background.

We devise a criterion based on *appearance* cues to avoid the ambiguity associated with merging regions based on motion. To this end, we use a pre-trained self-supervised image encoder, such as DINO-ViT [Caron et al. 2021], to obtain dense features for the input image and merge the segments predicted by Φ based on feature similarity. Formally, let V_u denote the feature vector of pixel u obtained by the self-supervised encoder. Then, $\bar{V}_k = \sum_u V_u p(m_u = k \mid I, \Phi) / \sum_v p(m_v = k \mid I, \Phi)$ is the average feature vector for segment k, where pixels are weighed by their probability with which they belong to the segment. We compute the pairwise similarities of different regions via an affinity matrix $\Pi \in \mathbb{R}^{K\times K}$, where entries corresponding to segments i and j are set as

$$(\Pi)_{ij} = \max\left(\epsilon, \left\langle \frac{\bar{V}_i}{||\bar{V}_i||_2}, \frac{\bar{V}_j}{||\bar{V}_j||_2} \right\rangle\right), \tag{3.6}$$

where only feature vectors pointing in the same direction are considered and $\epsilon = 10^{-12}$ is a small constant that keeps the graph connected. We then perform spectral clustering [Cheeger 1969; Jianbo Shi and Malik 2000; Melas-Kyriazi et al.

2022a] into two components using the affinity Π.

3.3.3 Two Scenarios: Motion *vs* Image Segmentation

We experiment with two modes of application of our model. The first scenario is *internal learning for unsupervised video segmentation*, where the network is evaluated on the same video sequences that have been used for optimization. This is effectively an unsupervised motion segmentation algorithm because the network not only receives as input appearance information, but incorporates motion information via backpropagation, observing indirectly optical flow too. While not explicitly stated in the respective papers, prior motion segmentation works such as [Charig Yang et al. 2021; Meunier et al. 2022] also operate in this mode, while directly observing moving objects, often using optical flow as input.

The second scenario is *transductive learning for image segmentation*. In this case, the network is first trained using a number of unlabelled videos, and then used for single-image foreground object segmentation on an *independent* validation/test set of still images. In this scenario, motion is only used as a supervisory signal: when the network is applied at test time, motion is not considered anymore and the network operates purely as an image-based segmenter. As for any transductive learning setting, the goal is to assess the generalization performance of the network on new images.

3.4 Experiments

As discussed above, our formulation allows us to evaluate our method in two settings: video object segmentation and general image/object segmentation. We show that learning a network that *guesses what moves* not only results in state-of-the-art performance in video segmentation, but also generalizes to image segmentation without further training.

3.4.1 Experimental Setup

Architecture. Our formulation enables us to use any standard image segmentation architecture for the model Φ. This has two main benefits: while training the model needs optical flow (and thus video data), inference can be performed

	Inf. RGB	Input Flow	Input Resolution	Flow Method	Runtime sec ↓	**DAVIS** $\mathcal{J}\uparrow$	**STv2** $\mathcal{J}\uparrow$	**FBMS** $\mathcal{J}\uparrow$
SAGE[W. Wang et al. 2018]	✓	✓	–	LDOF	0.9	42.6	57.6	61.2
NLC[Faktor and Irani 2014]	✓	✓	–	SIFTFlow	11	55.1	67.2	51.5
CUT[Keuper et al. 2015]	✓	✓	–	LDOF	103	55.2	54.3	57.2
FTS[Papazoglou and Ferrari 2013]	✓	✓	–	LDOF	0.5	55.8	47.8	47.7
CIS[Yanchao Yang et al. 2019]	✓	✓	192 × 384	PWCNet	0.1	59.2	45.6	36.8
AMD[R. Liu et al. 2021]	✓	✗	128 × 224	✗	–	57.8	57.0	47.5
MG[Charig Yang et al. 2021]	✗	✓	128 × 224	RAFT	0.012	68.3	58.6	53.1
EM[Meunier et al. 2022]	✗	✓	128 × 224	RAFT	–	69.3	55.5	57.8
OCLR[Xie et al. 2022]	✓	✓	480 × 832	RAFT	–	78.9	71.6	68.7
DS‡[Ye et al. 2022]	✓	✓	240 × 426	RAFT	1800 (22.5)‡	79.1	72.1	71.8
Ours (UNet)	✓	✗	128 × 224	RAFT	0.027	78.3	76.8	72.0
Ours (MaskFormer)	✓	✗	128 × 224	RAFT	0.059	**79.5**	**78.3**	**77.4**
CIS†[Yanchao Yang et al. 2019]	✓	✓	192 × 384	PWCNet	11	71.5	62.0	63.5
DyStaB†*[Yanchao Yang et al. 2021]	✓	✓	192 × 384	RAFT	–	80.0	74.2	73.2
Ours† (w/ CRF)	✓	✗	128 × 224	RAFT	3.73	**80.7**	**78.9**	**78.4**

Table 3.1: **Unsupervised video segmentation** on DAVIS2016, SegTrack-v2 (*STv2*), and FBMS59. † denotes the usage of CRFs and other extra significant post-processing (e.g., multi-step flow, multi-crop, temporal smoothing for CIS [Yanchao Yang et al. 2019]). ‡ DS is optimized per sequence; authors report 30 min training time for 80-frame video. * DyStaB utilises supervised pre-training.

on single images alone just like any image segmentation method. Second, using a standard architecture allows us to benefit from (self-)supervised pretraining, ensuring better convergence and broader generalization. We experiment with both convolutional and transformer-based architectures.

Datasets. For the *video segmentation* task, we use three popular datasets: DAVIS2016 (DAVIS) [Perazzi et al. 2016], SegTrackV2 (STv2) [F. Li et al. 2013], as well as FBMS [Ochs et al. 2014]. For the *image segmentation* task, we consider the Caltech-UCSD Birds-200 (CUB) dataset [Welinder et al. 2010] and three saliency detection benchmarks: DUTS [L. Wang et al. 2017], ECSSD [Jianping Shi et al. 2016], and DUT-OMRON [Chuan Yang et al. 2013].

Optical Flow. Our method derives its learning signal from optical flow. We estimate optical flow for all frames on DAVIS, STv2, and FBMS following the practice of MotionGrouping [Charig Yang et al. 2021]. We employ RAFT [Teed and J. Deng 2020] (supervised) using the original resolution for our main experiments. Please see the supplement for experiments with other flow methods.

Training Details. We use MaskFormer [B. Cheng et al. 2021] as our segmentation network, and use only the segmentation head. For the backbone and appearance features V, we leverage a ViT-B transformer, pre-trained on ImageNet [Russakovsky

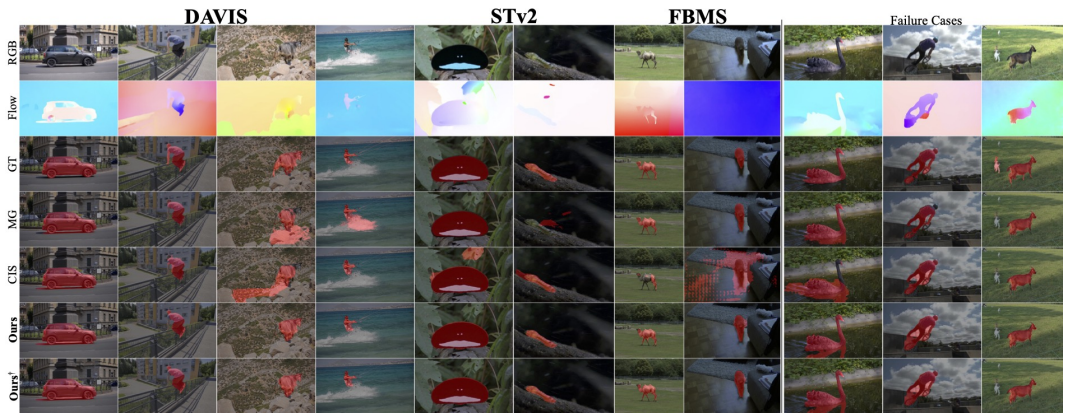


Figure 3.2: **Qualitative Comparison on DAVIS, STv2, and FBMS.** †– indicates use of CRF. Our method correctly segments objects in challenging conditions including strong parallax (2^{nd}, 3^{rd} *seq.*), small objects (4^{th}), background motion (5^{th}), camouflaged appearance (6^{th}), non-rigid motion (7^{th}) or no motion at all (8^{th} seq.). In the failure cases, our method is confused by ripples and reflection in the water, the front wheel rotating in a different direction and multiple disconnected objects.

et al. 2015] in a self-supervised manner using DINO [Caron et al. 2021] to avoid any external sources of supervision. We set the number of components to $K = 4$ unless otherwise noted. Please see the supplement for all details and hyper-parameter settings.

3.4.2 Unsupervised Video Segmentation

In Table 5.2 we report our performance on the DAVIS, STv2, and FBMS datasets and compare to other unsupervised video segmentation approaches. Our method achieves state-of-the-art performance, even without CRF post-processing. fig. 3.2 provides a qualitative comparison of the results. Our model provides better segmentation with sharper boundaries despite complex non-rigid motion, parallax effects or lack-of-motion. However, on challenging scenarios our method still struggles to segment small details or non-connected instances.

Our method is not restricted to a specific segmentation architecture. To investigate, MaskFormer is replaced with a simple convolutional U-Net architecture [Ronneberger et al. 2015], as in EM [Meunier et al. 2022], and trained from scratch for a fair comparison. The U-Net based model achieves comparable results on DAVIS and FBMS and 76.8 on STv2 (table 5.2), outperforming earlier methods even without transformers.

Flow Model	K	**DAVIS** ($\mathcal{J}\uparrow$)	**DAVIS** ($\mathcal{J}_{\text{oracle}}\uparrow$)
Constant ($A = 0$)	4	76.8	77.7
Affine ($u = [x,\ y]$)	4	77.1	78.8
Quadratic (eq. (3.2))	4	**79.5**	**81.5**
Quadratic (eq. (3.2))	2	74.5	74.5
Quadratic (eq. (3.2))	3	77.8	79.5
Quadratic (eq. (3.2))	4	**79.5**	**81.5**
Quadratic (eq. (3.2))	5	76.0	79.9

Table 3.2: **Flow Model and Number of Components.** We ablate the choice of flow model and the number of components K. More complex flow models improve performance, and over-segmentation helps until the assignment problem between components and the final binary segmentation becomes too difficult at $K = 5$. To evaluate the quality of the clustering of components we also report the oracle clustering performance as an upper bound.

3.4.3 Flow Model and Number of Components

Using DAVIS, we now study the effectiveness of the individual components of the method. In table 3.2 we evaluate the performance of the model under different flow models: constant, affine, and quadratic. We find that more complex models lead to improved performance, likely due to the fact that manny scenes in the DAVIS benchmark are highly dynamic with complex objects and backgrounds. Additionally, in the same table we evaluate how the number of components, K, influences the final performance after clustering. With $K = 2$ the model directly performs foreground-background separation but needs to model each with a single component which is often difficult, e.g. due to complex motions of deformable objects and/or parallax effects. Increasing the number of components is beneficial up to $K = 4$, after which the assignment problem from over-segmentation to foreground and background becomes too difficult for simple spectral clustering. This can be seen by evaluating the segmentation performance under an optimal oracle assignment of the components to foreground and background (oracle column in table 3.2). In all cases $K <= 4$, spectral clustering nearly reaches oracle performance.

3.4.4 Unsupervised Image Segmentation

While the main aim of our work is object segmentation in videos, we also assess the image segmentation performance on common image segmentation and saliency benchmarks: CUB, DUTS, DUT-OMRON, and ECSSD. For this experiment, we

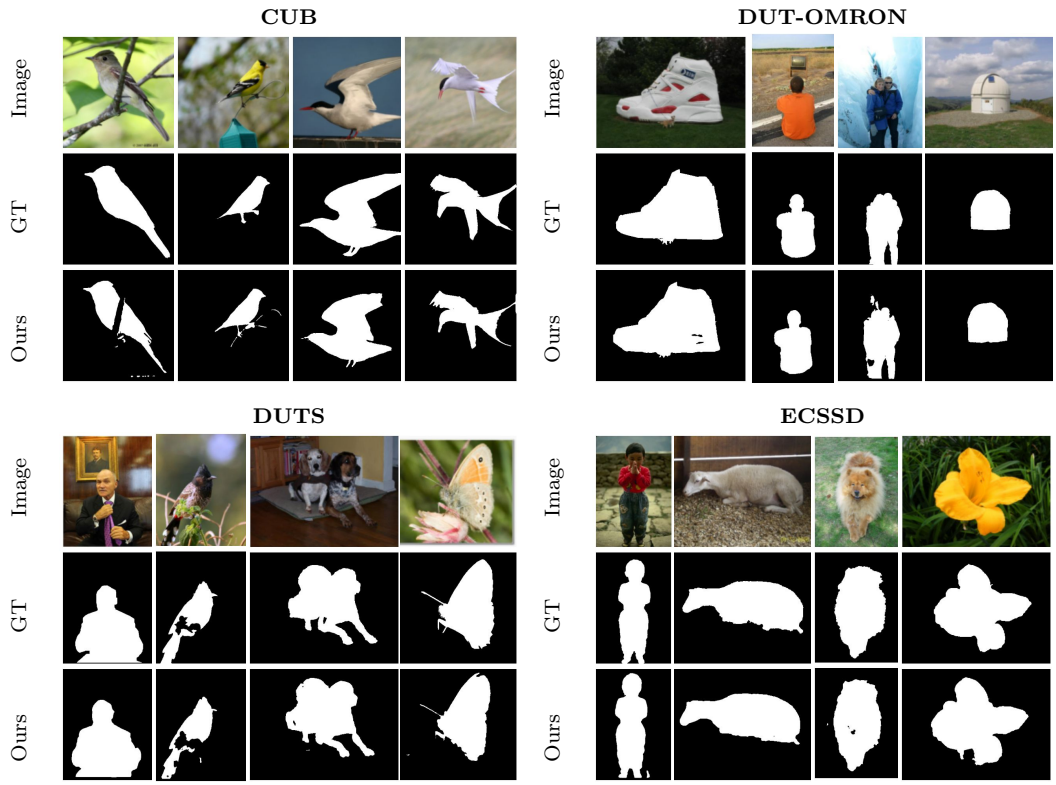


Figure 3.3: **Qualitative Comparison**. Our method can extract salient object in various environments and works even for novel object that were not included in the training data.

train our model on all three motion segmentation datasets (DAVIS, FBMS and STv2) jointly and apply the resulting network to the image segmentation benchmarks without any further fine-tuning. In table 3.3, we report the performance of our method and compare to the current state of the art. It is worth noting that most prior work (except [Melas-Kyriazi et al. 2022b; Melas-Kyriazi et al. 2022a; Y. Wang et al. 2022]) relies on dataset-specific training, self-training, post-processing or supervised pre-training to achieve image segmentation.

Finally, we evaluate the model qualitatively in fig. 3.3 on all four benchmarks. We observe our model works well on a diverse set of classes, such as buildings, certain animals and plants, even though they were not part of the foreground (moving) objects in the training data.

Limitations. Appendix C.3 contains further experiments and ablations. In appendix C.3.2, we show that lower quality optical flow estimation limits the performance of the segmenter. Quality of flow can be approximately measured using cycle consistency, which could be used as a weighting in the least-squares reconstruction of our loss to overcome the effect of poor quality flow.

	CUB			DUTS			ECSSD			OMRON		
	Acc	$\mathcal{J}\uparrow$	$maxF_\beta\uparrow$	Acc	$\mathcal{J}\uparrow$	$F_\beta\uparrow$	Acc	$\mathcal{J}\uparrow$	$F_\beta\uparrow$	Acc	$\mathcal{J}\uparrow$	$F_\beta\uparrow$
Voynov *et al.*[Voynov et al. 2021]	94.0	71.0	80.7	88.1	51.1	60.0	90.6	68.4	79.0	86.0	46.4	53.3
AMD[R. Liu et al. 2021]	–	–	–	–	–	60.2	–	–	–	–	–	–
Kyriazi *et al.*[Melas-Kyriazi et al. 2022b]	92.1	66.4	78.3	89.3	52.8	61.4	91.5	71.3	80.6	88.3	50.9	58.3
Kyriazi *et al.*[Melas-Kyriazi et al. 2022a]	–	76.9	–	–	51.4	–	–	73.3	–	–	56.7	–
DyStaB†[Yanchao Yang et al. 2021]	–	–	–	–	–	–	–	–	88.1	–	–	73.9
TokenCut[Y. Wang et al. 2022]	–	–	–	90.3	57.6	–	91.8	71.2	–	88.0	53.3	–
SelfMask[Shin et al. 2022a]	–	–	–	92.3	62.6	–	94.4	78.1	–	90.1	58.2	–
Ours	93.5	64.6	80.9	91.5	49.2	65.6	88.5	56.1	74.3	89.3	41.31	56.3

Table 3.3: **Unsupervised object segmentation** benchmark CUB and three saliency detection benchmarks: DUTS, ECSSD, and DUT-OMRON (*OMRON*). † DyStaB uses CRF post-processing, supervised pre-training, and self-training on each dataset. (SoTA only table - please see the supplement for a complete version of this table including many older methods.)

3.5 Conclusions

We have proposed a simple approach to exploit the synergies between motion in videos and objectness for segmenting visual objects without supervision. The key idea is using motion anticipation as a learning signal: we train an image segmentation network to predict regions that likely contain simple optical flow patterns, as these have a high chance to correspond to objects. We find that the complexity of the motion model is important to model complicated flow patters that can arise even for rigid objects. Our results show that this approach achieves state-of-the-art performance in video segmentation benchmarks. Future work could thus consider extensions to more sophisticated motion models, accounting for the 3D shape of objects, and to separate multiple objects.

Acknowledgments S. C. is supported by a scholarship sponsored by Facebook. L. K. is funded by EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems EP/S024050/1. I. L. and A. V. are supported by the European Research Council (ERC) grant IDIU-638009. I. L. is also supported by EPSRC grant VisualAI EP/T028572/1. C. R. is supported by Innovate UK (project 71653) on behalf of UK Research and Innovation (UKRI).

Statement of authorship can be found in appendix A. Supplementary material is available in appendix C.

Chapter 4

Unsupervised Multi-object Segmentation by Predicting Probable Motion Patterns

The Guess What Moves (GWM) method presented in the previous chapter was applied to binary segmentation. The loss in GWM can be lowered by splitting the object into multiple components to better approximate articulated motion and various depth discontinuities within the object. This was leveraged to support articulated motion by predicting multiple components and merging them using an appearance-based clustering. However, this is limiting when considering scenes with a variable number of objects, as some objects may be split up into multiple components without a way to merge them back.

In this chapter, we thus reconsider the problem of learning from motion in a multi-object setting. To this end, we approach the problem from a probabilistic perspective and consider the distribution over possible optical flow arrangements. We show how such an approach improves over previous methods in the multi-object setting. We also use the loss to complement other appearance-based appraoches, such as those benchmarked in chapter 2. Finally, we show how our loss enables learning segmentation of objects in real-world scenes containing multiple objects.

This chapter presents a paper that has been published in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Unsupervised Multi-object Segmentation by Predicting Probable Motion Patterns

Laurynas Karazija*, Subhabrata Choudhury*,
Iro Laina, Christian Rupprecht, Andrea Vedaldi
Visual Geometry Group
University of Oxford
Oxford, UK
{laurynas,subha,iro,chrisr,vedaldi}@robots.ox.ac.uk

Abstract

We propose a new approach to learn to segment multiple image objects without manual supervision. The method can extract objects form still images, but uses videos for supervision. While prior works have considered motion for segmentation, a key insight is that, while motion can be used to identify objects, not all objects are necessarily in motion: the absence of motion does not imply the absence of objects. Hence, our model learns to predict image regions that are likely to contain motion patterns characteristic of objects moving rigidly. It does not predict *specific* motion, which cannot be done unambiguously from a still image, but a distribution of possible motions, which includes the possibility that an object does not move at all. We demonstrate the advantage of this approach over its deterministic counterpart and show state-of-the-art unsupervised object segmentation performance on simulated and real-world benchmarks, surpassing methods that use motion even at test time. As our approach is applicable to variety of network architectures that segment the scenes, we also apply it to existing image reconstruction-based models showing drastic improvement. Project page and code: https://www.robots.ox.ac.uk/~vgg/research/ppmp.

*Authors contributed equally.

4.1 Introduction

Humans have an innate ability to segment individual objects in a picture, but learning this capability with an algorithm usually relies on manual supervision. In this paper, we consider the problem of learning to segment objects from visual data only — without externally provided labels. Algorithms for this task usually assume that objects are seen in different configurations and in front of different backgrounds. They then exploit cues such as the visual consistency and the co-occurrence of characteristic object parts to learn to discover and segment individual object instances.

Most such methods use still images as input and train with a reconstruction objective [Greff et al. 2019; Locatello et al. 2020; Jiang and Ahn 2020]. They work well on simple synthetic scenes, but they struggle in scenes with more complex visual appearance [Karazija et al. 2021]. This has motivated the development of algorithms that use videos as input and can thus observe the motion of the objects as evidence of their presence. A common way of using motion for unsupervised learning is to seek for a compact representation to *reconstruct* the video itself [Veerapaneni et al. 2020; Jiang et al. 2020; Z. Lin et al. 2020a; Kabra et al. 2021]. Effectively, such methods seek for a compressed representation of appearance, but do not sidestep entirely the difficult task of modeling it. This has motivated authors to look instead at reconstructing the video's *optical flow* [Charig Yang et al. 2021; Kipf et al. 2022]. In fact, the optical flow measures the motion of the objects directly and is much simpler to model than the objects' appearance.

In this paper, we propose a method that lies in-between these two classes, i.e. single image-based and video-based. Our model learns to segment objects in still images, and is thus based on appearance, but learns to do so using *video as a learning signal*, in an unsupervised manner. The learning process can be summarized as follows. Given an image, each pixel is assigned to a slot that represents a certain object. The quality of the assignments is then measured by the coherence of the (unobserved) optical flow within the extracted regions. Because predicting optical flow from a still image is intrinsically ambiguous, the method models *distributions* of probable flow patterns within each region. The idea is that (rigid) objects generate characteristic flow patterns that can be used to distinguish between them.

Note that, because the segmentation network is based on a single image, it will learn to partition all objects contained in the scene, not just the ones that actually move in the video, i.e. solving an object instance segmentation problem, rather than *motion* segmentation.

We derive closed-form distributions for the flow field generated by rigid objects moving in the scene. We also derive efficient expressions for the calculation of the flow probability under such models. The problem of decomposing the image into a number of regions is cast as a standard image segmentation task and an off-the-shelf neural network can be used for it.

As our method uses videos to train an image-based model, we introduce two new datasets which are straightforward video extensions of the existing image datasets CLEVR [Johnson et al. 2017] and ClevrTex [Karazija et al. 2021]. These datasets are built by animating the objects with initial velocities and using a physics simulation to generate realistic object movement. Our datasets are constructed with the realistic assumption that not all objects are moving at all times. This means that motion alone cannot be used as the sole cue for objectness and reflects scenarios such as a workbench where a person only interacts with a small number of objects at a time.

Empirically, we validate our model against several ablations and baselines. We compare our approach to existing unsupervised multi-object segmentation methods achieving state-of-the-art performance. We demonstrate particularly strong performance in visually complex scenes even with unseen objects and textures at test time. Our experiments in comparison to *image-based* models and, in particular, adding our motion-aware formulation to existing models shows substantial improvements, confirming that motion is an important cue to learn objectness. Furthermore, we show that our learned segmenter, which operates on still images, produces better segmentation results than current *video-based* methods that use motion information at test time. Finally, we also apply our method to real-world self-driving scenarios where we show superior performance to prior work.

4.2 Related Work

Multi-object decomposition. Learning unsupervised object segmentation for static scenes is a well-researched problem in computer vision [S. M. A. Eslami et al. 2016; Z. Lin et al. 2020b; Crawford and Pineau 2019; Jiang and Ahn 2020; C. P. Burgess et al. 2019; Emami et al. 2021; Engelcke et al. 2020; Engelcke et al. 2021; Locatello et al. 2020; Smirnov et al. 2021; Greff et al. 2016; Greff et al. 2015; Stelzner et al. 2019; K. Xu et al. 2019]. These methods aim to decompose the scene into constituent parts, e.g. the different foreground objects and the background. Glimpse-based methods [S. M. A. Eslami et al. 2016; Z. Lin et al. 2020b; Crawford and Pineau 2019; Jiang and Ahn 2020] find input patches (glimpses) that contain the objects in the scene. These methods learn object descriptors that encode their properties (e.g. position, number, size of the objects) using variational inference, composing glimpses into the final picture. More related to ours are approaches that learn per-pixel object masks [C. P. Burgess et al. 2019; Emami et al. 2021; Engelcke et al. 2020; Engelcke et al. 2021; Locatello et al. 2020]. MoNet [C. P. Burgess et al. 2019] and IODINE [Greff et al. 2019] employ multiple encoding-decoding steps to sequentially explain the scene as a collection of regions. Slot Attention [Locatello et al. 2020] uses a multi-step soft clustering-like scheme to find the regions simultaneously. In all cases, learning is posed as an image reconstruction problem. In order to align learnable slots with semantic objects, models have to make efficient use of a limited representation available for each region, such as learning to only explain visual appearance. This principle, however, is difficult to extend to visually complex data [Karazija et al. 2021] and relies on custom specialized architectures. Instead, our method allows for any standard segmentation architecture to be used, which we train to predict regions that are most likely described by rigid motion patterns.

Video-based multi object decomposition. Another line of work extends the unsupervised object decomposition problem to videos [Van Steenkiste et al. 2018; Kosiorek et al. 2018; Z. He et al. 2019; Weis et al. 2021; Zablotskaia et al. 2021; Kabra et al. 2021; N. Li et al. 2020b; Kossen et al. 2020; G. Singh et al. 2021; Y. Wu et al. 2021; Z. Lin et al. 2020a; Jiang et al. 2020]. Many of these methods work mainly with simpler datasets [Kossen et al. 2020; Zablotskaia et al. 2021; G. Singh et

al. 2021] and require sequential frames for training. For example, SCALOR [Jiang et al. 2020] is a glimpse-based method that discovers and propagates objects across frames to learn intermediate object latents. SIMONe [Kabra et al. 2021] processes the whole video at once, learning both temporal scene representation and time-invariant object representations simultaneously. Slot Attention for Video (SAVi) [Kipf et al. 2022] poses the multi-object problem as optical-flow prediction using sequential frames as input. The internal slot-attention mechanism drives the network to learn regions that move in a simple and consistent manner. Different to our work, it does not assume a specific motion model but relies on directly regressing the flow. It is computationally more expensive and struggles when only one or few frames are available.

Unsupervised video object segmentation. Unsupervised video object segmentation (VOS) is a popular problem in computer vision [Faktor and Irani 2014; Tsai et al. 2016; Papazoglou and Ferrari 2013; Tokmakov et al. 2019; Jain et al. 2017; Charig Yang et al. 2021; Yanchao Yang et al. 2019; S. Li et al. 2018; X. Lu et al. 2019], that focuses on extracting the most salient object in the scene. Many of the approaches treat the problem as a motion segmentation task as the background typically shows a dominant motion independent of the salient object. Motion Grouping [Charig Yang et al. 2021] employs the Slot Attention architecture to reconstruct optical flow from itself, avoiding appearance information entirely. In [H. Chen et al. 2022] objects are iteratively explained away starting from confident flow segments, which are refined using a graph propagation method. Another related line of work [Torr 1998; Mahendran et al. 2018; Meunier et al. 2022; Choudhury et al. 2022] employs approximate motion models. These approaches rely on a point estimate of the motion model parameters. In contrast, we adopt a more principled probabilistic approach, placing a prior on the motion parameters and integrating them out. To deal with flow outliers that do not conform to a rigid motion model, [Mahendran et al. 2018] use a histogram matching-based loss and GWM [Choudhury et al. 2022] over-segments the scene relying on spectral clustering to produce a binary segmentation during inference. Instead, we model the noise in our formulation directly. Finally, [Meunier et al. 2022] rely on flow as input, limiting the method to videos only.

4.3 Method

Let a frame $\mathcal{I} \in \mathbb{R}^{3\times H\times W}$ of a video and its optical flow $\mathbf{f} \in \mathbb{R}^{2\times H\times W}$ be defined on the $H\times W$ lattice. The optical flow is a local summary of the motion from one frame to the next. We use it to supervise a network Φ that, given the (single) image $\mathcal{I}$ as input, predicts soft assignments of each pixel to up to K different image regions, outputting an $H\times W$ collection of probability vectors $\Phi(\mathcal{I}) \in \hat{\Delta}_K^{H\times W} \subset [0,1]^{K\times H\times W}$, where $\hat{\Delta}_K$ is the $K-1$-dimensional simplex. The quality of the regions is measured based on how likely they contain *flow patterns* typical of the motion of independent objects.

In more detail, we represent the predicted image regions by a hard K-way pixel assignment (mask) $\mathbf{m} \in \Delta_K^{H\times W} \subset \{0,1\}^{K\times H\times W}$, where Δ_K is the space of K-dimensional one-hot vectors. Each mask is a sample from the categorical distribution output by the network, i.e. $\mathbf{m} \sim p_\Phi(\mathbf{m} \mid \mathcal{I}) = \text{Categorical}[\Phi(\mathcal{I})]$. Note that there is one categorical distribution for each pixel and that these are mutually independent.

We then assume that the flow depends only on the regions, in the sense that $p_\Phi(\mathbf{f}, \mathbf{m} \mid \mathcal{I}) = p(\mathbf{f} \mid \mathbf{m})\, p_\Phi(\mathbf{m} \mid \mathcal{I})$, where $p(\mathbf{f} \mid \mathbf{m})$ is a model of the distribution of the flow field given the regions. The likelihood of the modeled Φ is bounded by:

$$\log p_\Phi(\mathbf{f} \mid \mathcal{I}) = \log \mathop{\mathbb{E}}_{\mathbf{m}\sim p_\Phi(\mathbf{m}|\mathcal{I})} [p(\mathbf{f} \mid \mathbf{m})] \geq \mathop{\mathbb{E}}_{\mathbf{m}\sim p_\Phi(\mathbf{m}|\mathcal{I})} [\log p(\mathbf{f} \mid \mathbf{m})]\,.$$

Furthermore, inspired by ELBO, we regularize the model's prediction $p_\Phi(\mathbf{m} \mid \mathcal{I})$ by taking its KL divergence from a uniform prior $p_0(\mathbf{m})$, obtaining the learning objective

$$\mathcal{L}_\beta = \mathop{\mathbb{E}}_{\mathbf{m}\sim p_\Phi(\mathbf{m}|\mathcal{I})} [-\log p(\mathbf{f} \mid \mathbf{m})] + \beta D_{\text{KL}}\left(p_\Phi(\mathbf{m} \mid \mathcal{I}) \,\|\, p_0(\mathbf{m})\right). \tag{4.1}$$

Next, we introduce the closed-form motion model $p(\mathbf{f} \mid \mathbf{m})$ in eq. (4.1) and then explain how the Gumbel-Softmax trick can be used to train the network. The overall approach is summarized in fig. 6.2.

Approximate motion models for optical flow. We now turn to describing the models of motion used in our work, which play a role in assessing the likelihood of optical flow $p(\mathbf{f} \mid \mathbf{m})$. Optical flow measures the coordinate change of pixels

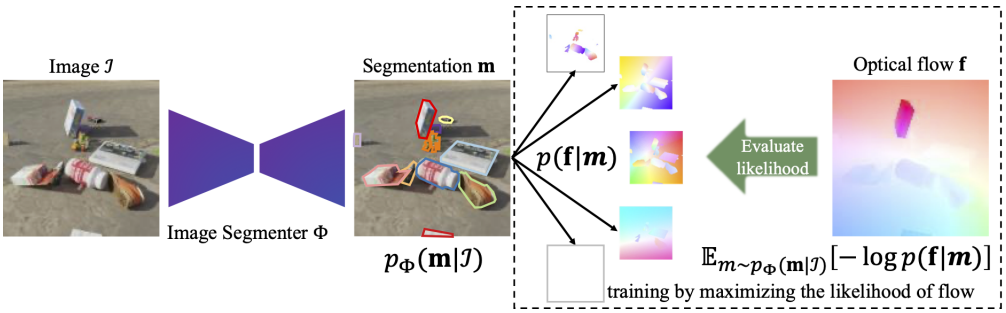


Figure 4.1: Overview of the proposed method. The segmentation network Φ is trained to predict a distribution over possible image regions. The flow model $p(\mathbf{f} \mid \mathbf{m})$ is used to assess the likelihood of the optical flow given the regions. The overall objective is to maximize the likelihood of the optical flow.

between neighboring frames, which arises due to the motion of the camera and objects. We consider rigid-body motion of some object k.

Let $\mathbf{x}_k^t, \mathbf{y}_k^t \in \mathbb{R}^{n_k}$ be the spatial locations of the pixels belonging to region/object k at time t, where n_k is the number of pixels in the region. For convenience, we stack the coordinates in a single vector $\Omega_k^t = (\mathbf{x}_k^t, \mathbf{y}_k^t) \in \mathbb{R}^{2n_k}$. The pixels comprising this object undergo coordinate change from Ω_k^t to Ω_k^{t+1}, giving rise to the optical flow for this object as $\mathbf{f}_k = \Omega_k^{t+1} - \Omega_k^t$. We assume this underlying 3D rigid-body motion can be approximated using a linear 2D parametric model Π_θ with parameters θ, so that:

$$\Omega_k^{t+1} = \Pi_\theta(\Omega_k^t) + \epsilon, \qquad \mathbf{f}_k = \Pi_\theta(\Omega_k^t) - \Omega_k^t + \epsilon, \tag{4.2}$$

where ϵ captures the residual error of the approximation. Several forms of models are available (see [Adiv 1985; Bergen et al. 1992] for an overview). Here, we consider two such models: the translation of an object within the camera plane, and an affine motion, given respectively by linear functions:

$$\Pi_\theta^{\text{tr}}(\Omega_k^t) = \Omega_k^t + \underbrace{\begin{bmatrix} \mathbf{1}_{n_k} & 0 \\ 0 & \mathbf{1}_{n_k} \end{bmatrix}}_{P_k^{\text{tr}}} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}, \quad \Pi_\theta^{\text{aff}}(\Omega_k^t) = \underbrace{\begin{bmatrix} \mathbf{x}_t & \mathbf{y}_t & \mathbf{1}_{n_k} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{x}_t & \mathbf{y}_t & \mathbf{1}_{n_k} \end{bmatrix}}_{P_k^{\text{aff}}} \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_6 \end{pmatrix}, \tag{4.3}$$

where we use $\mathbf{1}_{n_k}$ is a vector of n_k ones and matrix P_k contains the coefficients of the model.

The affine model supports object rotation, scaling and shearing in addition to translation. It is often a sufficient approximation to real-world optical flow, provided

the objects are rigid, convex, and mainly rotating in-plane.

We can then use the motion equations (4.2) to construct the distribution $p(\mathbf{f} \mid \mathbf{m})$ by assuming a prior on the motion parameters and by marginalizing over it. Specifically, denote by $\mathbf{m}_k$ the k-th slice of the tensor $\mathbf{m}$ encoding the regions (i.e. the mask of the k-th region). We assume that regions are statistically independent and decompose the log-likelihood $p(\mathbf{f} \mid \mathbf{m})$ as:

$$\log p(\mathbf{f} \mid \mathbf{m}) = \sum_k \log p(\mathbf{f}_k \mid \mathbf{m}_k) = \sum_k \int \log p(\mathbf{f}_k, \theta_k \mid \mathbf{m}_k) d\theta_k. \tag{4.4}$$

Assuming that each object has i.i.d. parameters θ_k with a Gaussian prior $\mathcal{N}(\theta; \mu, \Sigma)$, and assuming ϵ is a zero-mean noise with variance σ^2, eq. (4.2) gives marginal optical flow likelihood for segment k:

$$p(\mathbf{f}_k \mid \mathbf{m}_k) = \mathcal{N}\left(\mathbf{f}_k;\, \Pi_\mu(\Omega_k) - \Omega_k,\, P_k \Sigma P_k^\top + \sigma^2 I\right) \tag{4.5}$$

where I is the identity matrix. A practical issue with eq. (4.5) is that, if segment k contains $n_k = \sum_i (\mathbf{m}_k)_i$ pixels, then the covariance matrix $P_k \Sigma P_k^\top + \sigma^2 I$ has dimension $2n_k \times 2n_k$. Inverting such a matrix in the evaluation of the Gaussian log-density is very slow except for very small regions. Furthermore, it is not obvious how to relax eq. (4.4) to support gradient-based learning, e.g. through the Gumbel-Softmax approximation. We solve these problems in the next section.

Expressions for the likelihood. We now derive expressions for eq. (4.5) which are efficient and that lead to a natural relaxation for use in the Gumbel-Softmax sampling. Given the definitions $\mathbf{F}_k = \mathbf{f}_k - \Pi_\mu(\Omega_k) + \Omega_k$ and $\Lambda = \Sigma^{-1}$, we can rewrite eq. (4.5) as:

$$p(\mathbf{f}_k \mid \mathbf{m}_k) = (2\pi\sigma^2)^{-n_k} \left(\frac{\det S_k}{\det \Lambda}\right)^{-\frac{1}{2}} e^{-\frac{d^2}{2\sigma^2}}, \quad d^2 = \mathbf{F}_k^\top \mathbf{F}_k - \frac{1}{\sigma^2} \mathbf{F}_k^\top P_k S_k^{-1} P_k^\top \mathbf{F}_k \ , \tag{4.6}$$

where $S_k = {}^1\!/_{\sigma^2} P_k^\top P_k + \Lambda$. The significant advantage of this form is that it involves the computation of the inverse and determinant of matrix S_k, whose size is only 2×2 (for the translation model) or 6×6 (for the affine one), instead of the much larger $2n_k \times 2n_k$.

We can more explicitly introduce the dependency on the region assignments $\mathbf{m}$ by

defining selector matrices $R_k \in \{0,1\}^{2n_k \times 2n}$ (with $n = \sum_k n_k = HW$) that extract the $\mathbf{x}$ and $\mathbf{y}$ coordinates of the pixels that belong to the corresponding region, i.e. $\Omega_k = R_k\Omega$. We can then also write $\mathbf{F}_k = R_k\mathbf{F}$ and $P_k = R_kP$. Furthermore, the product of the selectors $L_k = R_k^\top R_k \in \{0,1\}^{2n\times 2n}$ can be written directly as a function of the assignment $\mathbf{m}$ as $L_k(\mathbf{m}) = \mathrm{diag}(\mathbf{m}_k, \mathbf{m}_k)$. Plugging these back in eq. (4.6), we obtain expressions involving L_k only:

$$n_k = \frac{1}{2}|L_k|_1, \quad S_k = \frac{1}{\sigma^2}P^\top L_k P + \Lambda, \quad d^2 = \mathbf{F}^\top L_k \mathbf{F} - \frac{1}{\sigma^2}(\mathbf{F}^\top L_k P) S_k^{-1} (P^\top L_k \mathbf{F}). \tag{4.7}$$

Translation-only model. Further simplifications are possible for specific models. For instance, for the translation-only model, assuming that $\Lambda = \mathrm{diag}(1/\tau^2, 1/\tau^2)$ then $S_k = \mathrm{diag}(n_k + 1/\tau^2, n_k + 1/\tau^2)$ and, after some calculations, we obtain the expression:

$$-\log p(\mathbf{f} \mid \mathbf{m}) = n \log 2\pi\sigma^2 + \sum_k \log \frac{n_k + \frac{\sigma^2}{\tau^2}}{\frac{\sigma^2}{\tau^2}} + \frac{1}{2\sigma^2}\mathbf{F}^\top \left(I - \sum_k \frac{1}{n_k + \frac{\sigma^2}{\tau^2}} \begin{bmatrix} \mathbf{m}_k\mathbf{m}_k^\top & \\ & \mathbf{m}_k\mathbf{m}_k^\top \end{bmatrix} \right) \mathbf{F}.$$

Affine model. For the affine model, the expression for $-\log p(\mathbf{f} \mid \mathbf{m})$ does not simplify as much. Still, by exploiting the structure of matrix P_k^{aff}, we can reduce the calculations to the computation of inverse and determinant of small 3×3 matrices, which can be implemented efficiently in closed form. Please see the Appendix for the derivation. Unless otherwise stated, the mean vector is set to $\mu = (1\ 0\ 0\ 0\ 1\ 0)^\top$ centering the prior on the no-motion point.

Gumbel-softmax. In order to train the network using gradient descent, we need a differentiable version of loss (4.1). To do so, we use the re-parametrizable Gumbel-softmax relaxation [Maddison et al. 2017; Jang et al. 2017]. The Gumbel-softmax relaxation replaces categorical samples $\mathbf{m} \in \Delta_K^{H\times W}$ from the distribution $p_\Phi(\mathbf{m} \mid \mathcal{I}) = \text{Categorical}[\Phi(\mathcal{I})]$ with continuous samples $\hat{\mathbf{m}} \in \hat{\Delta}_K^{H\times W}$ from the distribution $\text{GumbelSoftmax}[\Phi(\mathcal{I})]$. We take $N = 3$ samples from this distribution to evaluate the expected negative log-likelihood, further reducing variance. Then we simply replace $\hat{\mathbf{m}}$ for $\mathbf{m}$ in eq. (4.1), leading to differentiable quantities.

Post-processing. eq. (4.5) naturally encourages the model to form larger regions to explain parts of the scene that move in a consistent (under the assumed prior) manner. However, we find that this can also lead to the model grouping together objects that only coincidentally move together (e.g. all objects mostly falling due to gravity in one of the datasets). Furthermore, optical flow is ambiguous around object edges and occlusions. To address both the object grouping and occlusion boundary issue, we use a simple post-processing step. We isolate connected components in the model output, selecting the K largest masks, discarding any that are smaller than 0.1% of the image area, and combining the left-over and discarded ones with the largest mask overall.

Warp loss. Occasionally, the optical flow used to supervise our model can be noisy as it is estimated by other methods. This noise is also unlikely to be isotropic as some surfaces are easier to estimate that others. Rather than supporting heterogeneous noise and approximation error (eq. (4.2)), we instead prioritize parts of the scene covered by higher-quality flow. To this end, we introduce an additional loss term that simply enforces consistency between adjacent frames $\mathcal{I}_1, \mathcal{I}_2$. In particular, it warps the predicted mask distributions $\Phi(\mathcal{I}_1), \Phi(\mathcal{I}_2)$ using the optical flow, weighted by the error of warping the frames themselves, as follows:

$$\begin{aligned} \mathcal{L}_{\text{warp}}(\mathcal{I}_1, \mathcal{I}_2, f_1, b_2) &= w(\mathcal{I}_2, f_1(\mathcal{I}_1)) \cdot d(\Phi(\mathcal{I}_2), f_1(\Phi(\mathcal{I}_1))) \\ &\quad + w(\mathcal{I}_1, b_2(\mathcal{I}_2)) \cdot d(\Phi(\mathcal{I}_1), b_2(\Phi(\mathcal{I}_2))), \\ w(\mathcal{I}_a, \mathcal{I}_b) &= 1 - \text{norm}(|\mathcal{I}_a - \mathcal{I}_b|), \\ d(p, q) &= D_{\text{KL}}(p \parallel q)/2 + D_{\text{KL}}(q \parallel p)/2, \end{aligned} \tag{4.8}$$

where $f_1(\cdot)$ indicates warping by forward optical flow f_1 (or backward b_2). The symmetrized KL divergence, $d(\cdot)$, measured agreement between predicted and warped mask distributions, weighted by the absolute error of the warped frames normalized in $[0, 1]$. While the use of this term is not central to our method, we include it to show how tolerance to noisy optical flow can be improved. We do not use the warp loss (eq. (4.8)) in our experiments, unless otherwise indicated by (WL). In that case, the final loss is simply sum of the two terms: $\mathcal{L}_\beta + \mathcal{L}_{\text{warp}}$.

Figure 4.2: Example images and optical flow of the MovingCLEVR and MovingClevrTex datasets, which extend CLEVR and ClevrTex respectively to short videos based on physics simulation. Note that only a subset of objects is in motion in each frame.

4.4 Experiments

Our method lies in between image-based and video-based segmentation approaches, because it uses videos for supervision, but trains an image segmentation network that operates on still images only. We thus evaluate our approach under a number of settings. Firstly, we evaluate how well motion can be used to supervise an object instance segmenter that operates on still images. Secondly, we compare such a segmenter to state-of-the-art object segmentation methods that use motion also at test time (and are thus advantaged compared to our model). We conduct further analysis to validate our modelling assumptions and the model's reliance on the quality of the optical flow used for supervision. Finally, we apply our method to a real-world setting.

4.4.1 Experimental setup

Datasets. We evaluate our method on video and still image datasets. For video-based data, we use the Multi-Object Video (MOVi) datasets, released as part of Kubric [Greff et al. 2022]. Specifically, we employ MOVi-{A,C,D,E} versions. MOVi-A is similar to CLEVR [Johnson et al. 2017] in terms of visual complexity and contains videos of 3–10 falling objects on a simple, gray background. MOVi-C is significantly more challenging, as it features scanned, textured, common objects on top of backgrounds textured using HDR images. In MOVi-D, the number of objects is increased up to 23. In MOVi-E, the camera is additionaly moving. We use a resolution of 128×128 and the provided ground truth optical flow.

To evaluate our method on still images, we use CLEVR [Johnson et al. 2017] and ClevrTex [Karazija et al. 2021] benchmark suites. Both consist of images depicting 3–10 objects. CLEVR images are simpler with uniformly colored objects

with metallic or rubbery materials. CLEVRTEX features more diverse objects with complex textures applied. We also use the OOD and CAMO test sets from CLEVRTEX benchmark. OOD contains out-of-distribution shapes and textures. CAMO has camouflaged objects where the same texture is sampled for objects and the background.

Since our method requires optical flow during training, we extend the implementation of [Karazija et al. 2021] to generate video datasets of CLEVR and CLEVRTEX scenes, where a *subset* of objects contained in each scene are sliding, rolling and colliding based on a physics simulation (fig. 4.2). We generate 10k sequences for MOVINGCLEVRTEX and 5k for MOVINGCLEVR, where we retain 1000 and 500 sequences, respectively, for validation. Each sequence is 5 frames long. Dataset details can be found in the Appendix. The evaluation is performed on the original CLEVR and CLEVRTEX test sets.

We also evaluate our method on the real-world KITTI [Geiger et al. 2012] benchmark which depicts street scenes captured from a moving car. We follow the set up of [Bao et al. 2022], using 147 videos for training and evaluate on the instance segmentation subset which contains 200 annotated validation frames.

Metrics. Following prior work [Kipf et al. 2022; Karazija et al. 2021], we measure performance using two metrics. FG-ARI is the Adjusted Rand Index measured on foreground pixels only (selected using the ground-truth segmentation). Mean Intersection-over-Union (mIoU), is measured through Hungarian matching and averaged across the number of predicted or ground truth components, whichever is higher. When evaluating on videos, we calculate these metrics per-frame.

Network architecture. Our method can employ any image segmentation network architecture and train from scratch. Unless otherwise specified, we use Mask2Former [B. Cheng et al. 2022], using only its semantic segmentation. Following prior work [Locatello et al. 2020; Kipf et al. 2022], we use a 6-layer CNN backbone on synthetic datasets and ResNet-18 for KITTI. We also experiment with Swin-tiny transformer [Z. Liu et al. 2021] as the backbone. We use 11 transformer queries which become $K = 11$ slots on CLEVR, CLEVRTEX, and MOVi-A/C. On MOVi-D/E, we set $K = 24$ and $K = 22$ on KITTI. The model takes approximately

Table 4.1: Benchmark results on CLEVR, ClevrTex, CAMO, and OOD comparing FG-ARI and mIoU metrics (see also Appendix for an extended version). Results are a mean of 3 seeds ($\pm\sigma$). Methods above the line are trained on single images, while methods below train on videos.† – indicates post-processing.

	CLEVR		ClevrTex		OOD		CAMO	
Model	**FG-ARI↑**	**mIoU↑**	**FG-ARI↑**	**mIoU↑**	**FG-ARI↑**	**mIoU↑**	**FG-ARI↑**	**mIoU↑**
SPAIR [Crawford and Pineau 2019]	77.13± 1.92	65.95± 4.02	0.00± 0.00	0.00±0.00	0.00± 0.00	0.00±0.00	0.00± 0.00	0.00±0.00
SPAIR†	77.05± 1.96	66.87± 9.65	0.00± 0.00	0.00±0.00	0.00± 0.00	0.00±0.00	0.00± 0.00	0.00±0.00
MN [Smirnov et al. 2021]	72.12± 0.64	56.81±0.40	38.31± 0.70	10.46±0.10	37.29± 1.04	12.13±0.19	31.52± 0.87	8.79±0.15
MN†	72.08± 0.62	57.61±0.40	38.34± 0.73	10.34±0.12	37.28± 1.07	11.97±0.21	31.54± 0.87	8.77±0.18
MONet [C. P. Burgess et al. 2019]	54.47±11.41	30.66±14.87	36.66± 0.87	19.78±1.02	32.97± 1.00	19.30±0.37	12.44± 0.73	10.52±0.38
MONet†	61.36± 7.33	45.61± 4.80	35.64± 1.17	23.59±0.29	31.51± 1.46	23.04±0.52	9.94± 0.50	11.31±0.30
SA [Locatello et al. 2020]	95.89± 2.37	36.61±24.83	62.40± 2.23	22.58±2.07	58.45± 1.87	20.98±1.59	57.54± 1.01	19.83±1.41
SA†	94.88± 1.67	37.68±26.56	61.60± 2.29	21.96±1.79	57.41± 1.92	20.60±1.45	56.85± 1.12	19.42±1.42
IODINE [Greff et al. 2019]	93.81± 0.76	45.14±17.85	59.52± 2.20	29.17±0.75	53.20± 2.55	26.28±0.85	36.31± 2.57	17.52±0.75
IODINE†	93.68± 0.83	44.20±18.67	60.63± 2.50	29.40±1.10	54.92± 2.24	27.96±0.81	38.29± 1.40	18.87±0.52
DTI-S [Monnier et al. 2021]	89.54± 1.44	48.74± 2.17	79.90± 1.37	33.79±1.30	73.67± 0.98	32.55±1.08	72.90± 1.89	27.54±1.55
DTI-S†	89.86± 1.78	53.38± 3.51	79.86± 1.36	32.20±1.49	73.60± 0.97	30.74±1.22	72.89± 1.88	26.30±1.57
GNM [Jiang and Ahn 2020]	65.05± 4.19	59.92± 3.72	53.37± 0.67	42.25±0.18	48.43± 0.86	40.84±0.30	15.73± 0.89	17.56±0.74
GNM†	65.67± 4.23	63.38± 3.76	53.38± 0.67	44.30±0.19	48.44± 0.86	42.87±0.28	15.72± 0.89	18.53±0.75
SAVi [Kipf et al. 2022]	—	—	49.54	31.88	42.68	30.31	42.67	29.60
Ours	91.69± 0.30	66.70± 0.32	90.80± 0.22	55.07±0.44	76.01± 0.56	46.84±0.20	72.78± 1.31	42.30±1.09
Ours †	**95.94**± 0.43	**84.86**± 4.06	**92.61**± 0.22	**77.67**±0.25	**78.24**± 0.43	**55.54**±0.44	**77.43**± 0.86	**56.43**±0.80

48h to train on a single A30 24GB GPU.[1] All training details and hyper-parameters are included in the Appendix.

4.4.2 Unsupervised multi-object segmentation in images

In table B.1, we evaluate our method on the CLEVR [Johnson et al. 2017] and ClevrTex [Karazija et al. 2021] benchmarks and compare to prior work. Our method outperforms image models based on appearance reconstruction on both metrics (mIoU and FG-ARI) and across all datasets. The performance gap increases on the visually complex ClevrTex, OOD, and CAMO variants, demonstrating the strong inductive bias that motion provides during training, especially when the objects are camouflaged. Note that, in this setting, our model is advantaged compared to the other models in table B.1, as it can observe (through the loss) the optical flow of the training scenes. For this reason, we also train the optical flow-based, unconditional SAVi [Kipf et al. 2022] model. Nevertheless, we find that despite having access to motion information during training, SAVi does not surpass appearance-only models, likely due to only having access to single frames at test time.

Post-processing helps improve results further. As shown in fig. 4.3, separating connected components in post-processing distinguishes objects that might be assigned to the same mask and suppresses boundary segments that tend to group

[1]Approx. total compute in this paper: 100 GPU days for our models, 154 GPU days for comparisons.

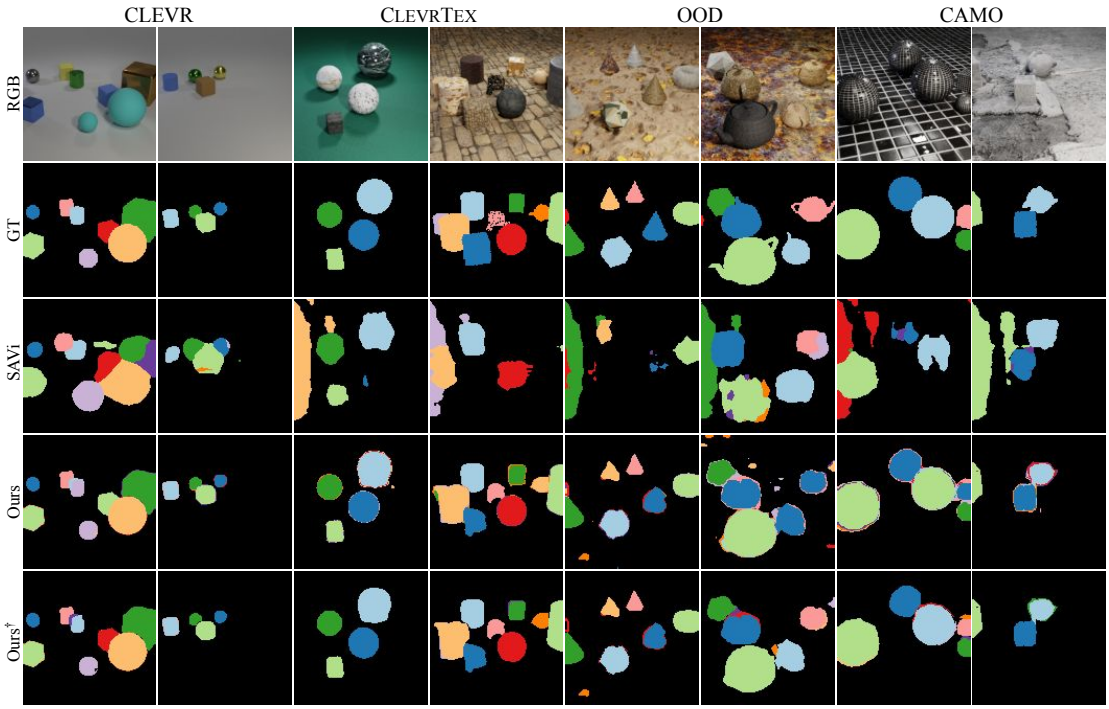


Figure 4.3: Unsupervised object segmentation on CLEVR and CLEVRTEX benchmarks. Our model is able to segment simple and visually complex scenes. Occasional mistakes around object boundaries and the assignment of different objects to the same component are addressed by post-processing. † – indicates post-processing.

Table 4.2: Segmentation results on MOVi datasets. Mean ± standard error (5 seeds). We calculate metric for each frame. All values in %. (WL) marks use of warp loss. † – indicates post-processing.

	MOVi-A		MOVi-C		MOVi-D		MOVi-E	
Model	**FG-ARI↑**	**mIoU↑**	**FG-ARI↑**	**mIoU↑**	**FG-ARI↑**	**mIoU↑**	**FG-ARI↑**	**mIoU↑**
GWM [Choudhury et al. 2022]	70.30	42.27	49.98	30.17	39.78	18.38	42.50	18.74
SCALOR [Jiang et al. 2020]	59.57	44.41	40.43	22.54	–	–	–	–
SAVi [Kipf et al. 2022]	**88.30**	62.69	43.26	31.92	43.45	10.60	17.39	5.75
Ours	84.01±0.72	60.08±1.47	61.18±0.84	34.72±0.17	**55.74**±1.02	23.50±0.35	62.62±0.92	25.78±0.27
Ours†	85.41±1.00	**76.19**±2.05	**61.24**±0.85	**37.26**±0.33	55.18±0.94	**25.21**±0.29	**63.11**±0.91	**28.59**±0.29
Ours† (Swin + WL)	90.08	84.76	67.64	52.17	66.41	30.40	72.73	35.30

difficult occlusion boundaries. For a fairer comparison, we also test if this post-processing improves the results of other methods but only obtain mixed results.

4.4.3 Unsupervised multi-object segmentation in video

We now evaluate our approach on video segmentation, where motion is available at test time. We report the performance of our method in Table 4.2 compared to video-based models: SCALOR [Jiang et al. 2020], *unconditional* SAVi [Kipf et al. 2022], and GWM [Choudhury et al. 2022]; the latter two also use optical flow supervision. It is important to note that SAVi and SCALOR make predictions jointly over all frames of a video, that allows them to actually see the objects in

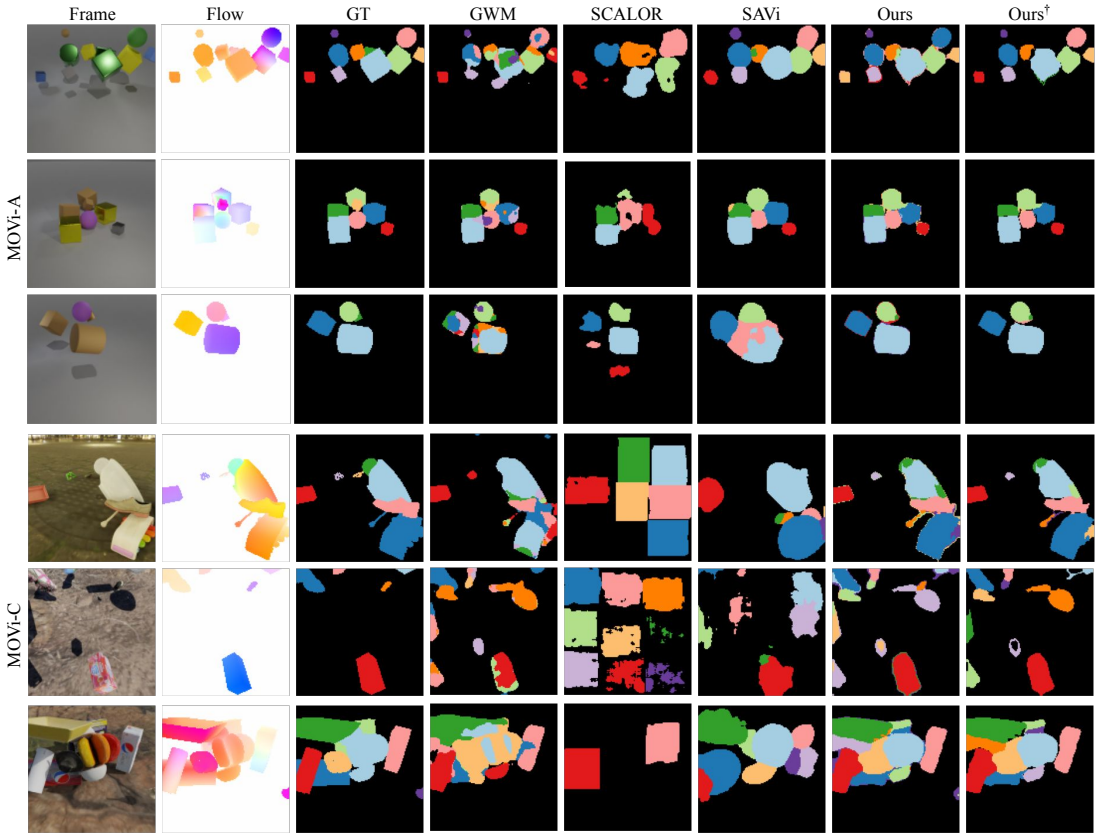


Figure 4.4: Qualitative comparisons on MOVi-A and MOVi-C. Our method performs consistently well compared to other methods. GWM suffers from oversegmentation where SCALOR has undersegmentation issue. Among the related methods, SCALOR fails to discover all the objects and SAVi's object boundaries are coarser †– indicates post-processing.

motion at test time. Despite this comparison being unfair to our approach, which operates on a single frame at a time, we achieve competitive results. On the visually simpler MOVi-A, our method has more than 10% lead over SAVi in mIoU, but performs slightly worse in terms of FG-ARI. On the more complex MoVi-C/D/E datasets our model shows strong performance, outperforming prior work on both metrics, with the performance gap again increasing with data complexity. Finally, we experiment with a version of the model using a deeper backbone (Swin) and the warp loss (eq. (4.8)). Though not necessary to achieve state-of-art results, this drastically improves performance on all metrics and dataset versions. In fig. 4.4, we also compare these models qualitatively, demonstrating that the different objects are overall better captured by our model with more refined boundaries, which explains the higher mIoU.

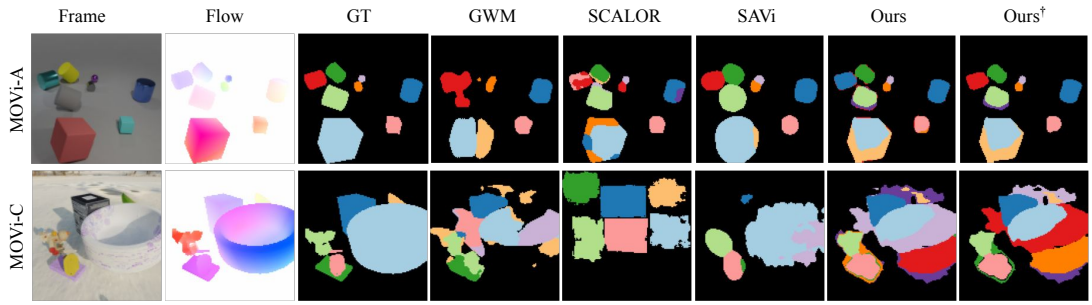


Figure 4.5: Failure cases on MOVi-A and MOVi-C. Our method has difficulty with objects that typically exhibit complex motion. It results in an over-segmentation of the object. Due to inherently imprecise optical flow near the boundaries our method also has a tendency to segment boundary pixels into a separate mask, which we fix with our post-processing step. †– indicates post-processing.

Table 4.3: Model ablations. (4.3a) replacing flow supervision with an unsupervised flow method, (4.3b) compares a translation-only with the affine flow model, and (4.3c) adds our motion-based formulation to an appearance-only model. All models with post-processing applied.

(a) Choice of Optical Flow Method

	MOVi-A		MOVi-C	
Optical Flow	**FG-ARI↑**	**mIoU↑**	**FG-ARI↑**	**mIoU↑**
SMURF [Stone et al. 2021]	80.17	26.3	61.21	28.77
Ground Truth	83.48	72.61	58.59	35.67

(b) Choice of Motion Model

	MOVi-A		MOVi-C	
Motion Mdl.	**FG-ARI↑**	**mIoU↑**	**FG-ARI↑**	**mIoU↑**
Translation	66.03	59.94	39.77	32.23
Affine	83.48	72.61	58.59	35.67

(c) Adding motion awareness to appearance-only models

		ClevrTex		OOD		CAMO	
Model	**Train data**	**FG-ARI↑**	**mIoU↑**	**FG-ARI↑**	**mIoU↑**	**FG-ARI↑**	**mIoU↑**
GNM [Jiang and Ahn 2020]	ClevrTex	53.38	44.30	48.44	42.87	15.72	18.53
GNM [Jiang and Ahn 2020]	MovingClevrTex	18.01	31.47	15.57	15.57	0.21	14.68
GNM+Our Loss	MovingClevrTex	63.84	55.26	59.01	48.65	51.00	47.63
SA [Locatello et al. 2020]	ClevrTex	62.40	22.58	58.45	20.98	57.54	19.83
SA [Locatello et al. 2020]	MovingClevrTex	61.84	21.44	58.24	20.67	57.30	18.82
SA+Our Loss	MovingClevrTex	76.60	38.12	67.01	33.95	70.59	33.05

4.4.4 Model ablations

Optical flow. In table 4.3a, we replace the ground-truth optical flow used so far in our experiments with the one estimated by SMURF [Stone et al. 2021]. The additional noise impacts our model's accuracy, as evidenced by the significant drop in mIoU (but comparable FG-ARI scores).

Motion model. In table 4.3b, we compare our affine motion model to the simpler translation-only one. We observe that the ability to describe complex motion patterns with an affine model improves performance over the translation model, which can only represent translation in the camera plane.

Motion awareness. Finally, we investigate the effectiveness of our objective in combination with existing appearance-based methods. The goal of this experiment

is to understand the advantage of using motion information during training (if available) and to decouple the effect of our formulation from the choice of architecture. To this end, we employ our objective on top of two models based on appearance reconstruction, GNM [Jiang and Ahn 2020] and SA [Locatello et al. 2020], with no other modifications. We train GNM and SA with and without our loss on videos (MovingClevrTex) and evaluate on the corresponding single-image test sets of the ClevrTex suite. In table 4.3c, we compare these models respectively to the original methods trained on static images (ClevrTex). We find that when trained on video data (MovingClevrTex), without our loss, both GNM and SA struggle. We attribute this to the reduced number of scenes in MovingClevrTex compared to ClevrTex. However, we note that using our loss significantly improves the performance of the appearance methods, suggesting the effectiveness of exploiting motion information through our formulation.

4.4.5 Segmentation on real-world data

We now turn to assessing our model's performance in a real-world setting. We follow the setting of [Bao et al. 2022] and evaluate on KITTI [Geiger et al. 2012], using RAFT [Teed and J. Deng 2020] to estimate optical flow and ResNet-18 as the backbone of our model, trained from scratch. We lower the input resolution for our model from $368 \times 1,248$ [Bao et al. 2022] to 288×960, which enables us to fit on a single GPU. We evaluate at 96×320 resolution.

As we show in table 4.4, our method outperforms prior work on the challenging real-world setting. In the same table, we also consider our method with an additional warp loss term, which further boosts performance. We also experiment with a transformer-based backbone (Swin) which is also pre-trained using self-supervision. Although, not necessary to show state-of-art result, this significantly improves real-world performance.

Table 4.4: Real-world segmentation results on KITTI. Baseline results from [Bao et al. 2022]. [Bao et al. 2022] and our method use RAFT for optical flow. Models above the line use ResNet-18 backbone. (WL) marks use of warp loss.

	KITTI
Model	**FG-ARI↑**
SA [Locatello et al. 2020]	13.8
MONet [C. P. Burgess et al. 2019]	14.9
SCALOR [Liang Liu et al. 2020]	21.1
S-IODINE [Greff et al. 2019]	14.4
MCG [Arbelaez et al. 2014]	40.9
[Bao et al. 2022]	47.1
Ours	50.8
Ours (WL)	**51.9**
Ours (Swin + WL)	58.3

4.4.6 Limitations

Motion is sometimes insufficient to distinguish different objects, for instance because they do not move or because they move similarly. In principle, this should not matter if sufficient motion diversity is observed in the training data as a whole; in practice, our model occasionally merges different object at test time, which we address partly in post-processing. Further improvements could be obtained by choosing a more informative prior $p_0(\mathbf{m})$ in eq. (4.1), to capture other desirable properties of objects, such as compactness and connectedness.

Figure 4.5 shows some failure cases where the affine motion model struggles to capture strong perspective effects caused by non-smooth depth changes in the object geometry. This could be addressed by modeling the object geometry (depth) and the ensuing complex flow patterns. Alternatively, this could be dealt with using a hierarchical segmentation model that can account for geometric discontinuities and self-occlusions. Hierarchical segmentation would also help with pronounced non-rigid motion (e.g. humans dancing or animals running) as motion could be explained at the level of object parts. While we approach a more general setting of multi-object segmentation, the proposed method does not offer any additional benefits in the limited cases of binary segmentation, such as those explored in [Choudhury et al. 2022]. We believe that incorporating a quadratic approximate motion model and hierarchical segmentation would be required to match the performance of the specialised models, such as GWM [Choudhury et al. 2022], for the binary case.

4.5 Conclusions

We have presented a method that bridges the gap between image-based and video-based scene decomposition, in that it requires only a single image as input, yet exploits motion cues available in videos during training. In comparison to prior work on image-based multi-object segmentation, our approach shows that motion provides useful objectness cues, especially as the visual complexity of a scene increases. Different from video-based approaches, however, our model operates on still images and does not rely on motion to detect or refine objects, which makes it more generally applicable. Finally, we deviate from the common objective of

image or flow *reconstruction* and, instead, model the problem by only predicting regions likely to contain affine flow patterns. This does not require a specialized architecture, thus any segmentation network is suitable for this task. Our approach achieves state-of-the-art performance on multiple image and video benchmarks, in simulated and real-world settings, validating the paradigm of training image models using motion.

Broader impact. Our work introduces a principled method for unsupervised multi-object segmentation. The work is mainly evaluated on 3D simulated datasets that do not contain people or personal information. Additionally, we evaluate on KITTI, a real-world self-driving dataset, which occasionally contains images of pedestrians. Consent cannot be obtained in this case, but we follow the KITTI terms of usage. We build on top of open source projects, respecting licenses and release all code, trained models and datasets for research purposes. Currently, the application of this approach is mainly limited to simulated imagery. Although the results on KITTI show promise, the immediate broader impact of our work in real-world scenarios, beyond the research community, is limited.

Acknowledgements L. K. is funded by EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems EP/S024050/1. S. C. is supported by a scholarship sponsored by Facebook. I. L., C. R. and A. V. are supported by European Research Council (ERC) grant 2020-CoG-101001212 UNION. I. L. and C. R. are also funded by EPSRC grant VisualAI EP/T028572/1.

Statement of authorship can be found in appendix A. Supplementary material is available in appendix D.

Chapter 5

Learning segmentation from point trajectories

In this chapter, we build on the ideas of previous chapters of learning objectness from motion. We consider the problem of long-term motion, which can capture intricate and unique motion patterns. We explore point trajectories as a source of such long-term motion information and use them to supervise a segmentation network in lieu of annotations. To enable this, we develop a loss function that measures whether point trajectories within masks correlate well-enought to belong to a single object. Our experiments shows that our loss and point trajectories complement optical flow-based loss and further improve performance in unsupervised video segmentation.

This chapter presents a paper that has been published in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. Additionally, it has been awarded a Spotlight presentation.

Learning segmentation from point trajectories

Laurynas Karazija, Iro Laina, Christian Rupprecht, Andrea Vedaldi

Visual Geometry Group

University of Oxford

Oxford, UK

{laurynas,iro,chrisr,vedaldi}@robots.ox.ac.uk

Abstract

We consider the problem of segmenting objects in videos based on their motion and no other forms of supervision. Prior work has often approached this problem by using the principle of common fate, namely the fact that the motion of points that belong to the same object is strongly correlated. However, most authors have only considered instantaneous motion from optical flow. In this work, we present a way to train a segmentation network using long-term point trajectories as a supervisory signal to complement optical flow. The key difficulty is that long-term motion, unlike instantaneous motion, is difficult to model – any parametric approximation is unlikely to capture complex motion patterns over long periods of time. We instead draw inspiration from subspace clustering approaches, proposing a loss function that seeks to group the trajectories into low-rank matrices where the motion of object points can be approximately explained as a linear combination of other point tracks. Our method outperforms the prior art on motion-based segmentation, which shows the utility of long-term motion and the effectiveness of our formulation.

5.1 Introduction

Segmentation, the task of delineating and isolating distinct objects, is a fundamental problem in computer vision. Much of the current approaches are supervised, relying on expensive manual annotations. Attempts to approach this task without supervision have largely relied on manual heuristics or exploited the rich semantics of self-supervised feature extractors. Video data, however, offers an additional option as it contains *motion*, which can be exploited for an additional inductive bias. Such approaches are rooted in the principle of common fate from Gestalt psychology [Wertheimer 1912], which posits that elements that move together are more likely to belong together.

Motion information is usually captured by optical flow. Flow is attractive as it arises from low-level visual properties and can provide a signal before scenes are parsed and objects are discovered. Furthermore, optical flow estimators, such as RAFT [Teed and J. Deng 2020] or FlowFormer [Zhaoyang Huang et al. 2022], can be trained purely on synthetic artificial data, transferring to real-world scenes with remarkable accuracy and without manual annotation. This has led many to consider optical flow as a critical modality to discover and learn objects from video data by learning to attribute and explain the motions of objects.

Optical flow, however, only describes the instantaneous motion of the scene, which can create blindspots: not all objects are necessarily in motion at all times. Similarly, groups of objects might coincidentally move together. Recent advances in point tracking [Karaev et al. 2024; Doersch et al. 2023; Doersch et al. 2022; Harley et al. 2022] offer an alternative form of motion information. Point trackers "lock on" to a set of query points and describe their position and visibility over the course of the whole video. This provides long-term motion information. Like optical flow estimators, point trackers are trained on synthetic data. However, unlike optical flow, point trajectories describe only a sparse set of points.

In this paper, we ask whether the long-term motion information obtained in point trajectories is beneficial. To that end, we explore how to supervise image segmentation networks using motion information with point trajectories. At a glance, this presents several problems. Firstly, point trajectories are time-varying 2D point clouds, and combining them with image-based networks is not straightforward.

Furthermore, the evolution of long-term object motion is too complex, even in the simplest cases. Our main insight is that the motion of points belonging to the same object should be well correlated. We thus propose a loss function that encodes this intuition by seeking to explain groups of points as combinations of other points in the group. With our method, a segmentation network predicts objects in the scene, inducing a grouping of trajectories that are currently visible. The loss function then assesses how well such a grouping explains the long-term motion. While point trajectories describe motion over a longer time, they are limited by the number of points tracked, which is often much less than the number of pixels. We thus propose to train using both trajectory-based loss and optical flow-based loss and show that spatially sparse but longer-time motion information synergises with spatially dense optical flow.

Discovering objects using point trajectories has a long history in computer vision. Our approach is inspired by ideas of subspace clustering, which assume that data comes from distinct subspaces and seek to reconstruct membership information of data points. This has previously been applied to the problem of motion segmentation [G. Liu et al. 2012; Elhamifar and Vidal 2013]. These approaches, however, are sensitive to noise and either rely on specialised optimisation procedures to recover a graph of trajectory relationships [Ochs and Brox 2011; Keuper et al. 2016] or use manual instead [Ochs and Brox 2011; Keuper et al. 2016]. Normalised cuts or spectral clustering are then used to group the trajectories. However, the need for an affinity matrix limits the number of trajectories that can be used due to quadratic memory requirements. Furthermore, "densification" is still required to extend trajectory clusters to the whole image. By construction, these approaches can process only a single sequence at a time. Our proposal instead trains an image segmentation network directly end-to-end using a dataset of videos while supporting a large number of trajectories.

In summary, our work makes the following contributions. (1) We propose a loss function that enables training any image segmentation architecture using point trajectories as a source of supervision. (2) We investigate our proposed loss in a principled way in a simulated setting, showing the feasibility of our approach. (3) We apply such a loss in a per-sequence optimisation, outperforming other subspace clustering baselines. (4) We use our loss to train a single network on a dataset

of videos for the task of video object segmentation, demonstrating strong results. (5) We show how our proposed loss formulation obtains better performance than alternatives.

5.2 Related work

Unsupervised video object segmentation. Video object segmentation (VOS) aims to label pixels of objects in a video. Current VOS benchmarks [Perazzi et al. 2016; F. Li et al. 2013; Ochs et al. 2014] usually define the problem as binary foreground-background separation or salient object segmentation. The task is usually approached in two ways: semi-supervised and unsupervised VOS. Semi-supervised methods require initial frame annotations and aim to *propagate* them to the rest of the video [Caelles et al. 2017]. Unsupervised VOS aims to discover object(s) of interest without the initial targets [Faktor and Irani 2014; Papazoglou and Ferrari 2013; Tokmakov et al. 2019; Jain et al. 2017; S. Li et al. 2018; X. Lu et al. 2019]. This however does not differentiate methods based on data used to *train* them. Most of the traditional research in semi- or unsupervised VOS relies on annotations during training. Our approach, in contrast, does not rely on any manual annotations to learn. Some authors explore related unsupervised video instance segmentation [Xudong Wang et al. 2023b] task without any annotations, object-centric learning appraoches [G. Singh et al. 2022b; Zadaianchuk et al. 2024; Aydemir et al. 2024], some of which make use of flow [Karazija et al. 2022] and depth [Safadoust and Güney 2023].

Motion segmentation. A closely related task to video object segmentation is motion segmentation, which aims to extract the main moving objects in a video. The practical difference between these two tasks is more difficult to delineate as the same benchmark datasets are often used. Early works modeled the scenes as layers [Chang and III 2013; Jojic and Frey 2001], which later works accomplish using a slot-attention mechanism [Charig Yang et al. 2021; S. Ding et al. 2022; Lao et al. 2023]. Flow mixture models accounted for multiple motion patterns [Jepson and Black 1993; Torr 1998], and corrections were introduced for rotating cameras [Bideau and Learned-Miller 2016; Bideau et al. 2018]. Later works [Meunier et al. 2022; Meunier and Bouthemy 2023a; Choudhury et al. 2022] considered parametric flow

models fit to explain the scene. AMD [R. Liu et al. 2021] employs a single model with separate appearance and motion 'pathways'. Other works train flow-only models by generating synthetic data, which generalise well to real videos [Xie et al. 2022; Lamdouar et al. 2021]. An alternative line of work adopts a more generative approach, training an inpainter networks to predict optical flow [Yanchao Yang et al. 2019; Yanchao Yang et al. 2021]. Several authors [Lian et al. 2023; S. Singh et al. 2023] adopt a multi-stage self-labelling [Xudong Wang et al. 2023b] approach for motion segmentation: initial masks are estimated using an optical flow-based approach, followed by DINO-based refinement and CRF post-processing to generate pseudo-labels and train a final segmentation network.

Trajectory-based motion segmentation. Trajectory-based motion segmentation has also been explored. Older works consider data of multiple trajectories and employ non-negative matrix factorization and related decomposition methods [Cheriyadat and Radke 2009; J. Costeira and Kanade 1995; Elhamifar and Vidal 2013; Fradet et al. 2009; Rao et al. 2008; Yan and Pollefeys 2006]. This line of work primarily operates by defining affinity between pairwise trajectories in a single video setting. In [Brox and Malik 2010b; Ochs and Brox 2012; Ochs et al. 2014; Keuper et al. 2015; Keuper 2017], heuristic graphs are constructed between trajectories, considering increasingly complex motion models, and employing specialised solvers to solve the optimisation problem. However, due to the specialised optimisation procedures and tight coupling with trajectory estimation methods, this line of work has received less attention than deep methods that exploit optical flow similarly to RGB frames.

Subspace clustering. A specific kind of trajectory-based technique is subspace clustering approaches, which rely on the *self-expressive* property of the data. They can largely be summarised [Haeffele et al. 2020] as solving a constrained optimisation problem $main : \#1 \min_C ||DC - D||_F^2 + \lambda\theta(C)$ for some dataset $D \in \mathbb{R}^{d \times n}$ of n points in d dimensions. C is a matrix of coefficients, which expresses the data and can be represented as a linear combination of other points. Given a solution for the coefficient matrix, it is transformed into an affinity matrix for spectral clustering. The approaches mainly differ in the second term of the objective and specialized methods to solve the optimisation problem. SSC [Elhamifar and Vidal

2013] define $\theta(C)$ as l_1 norm. LLR [G. Liu et al. 2012] use nuclear norm instead, while LSR [C.-Y. Lu et al. 2012] uses instead l_2 regularisation. [C.-Y. Lu et al. 2012; D. Luo et al. 2011; Vidal and Favaro 2014] combines l_1, l_2, and nuclear norms. Under some strong assumptions [Haeffele et al. 2020], these approaches enjoy some theoretical guarantees. However, they are difficult to scale in practice as the number of points n grows, as C is $n \times n$. Additionally, the secondary step of spectral clustering is also limiting and difficult to tune. Instead, we take inspiration from these approaches and propose a way to supervise the network directly using the self-expressive property of point trajectories.

5.3 Method

Our goal is to solve the video segmentation task in an unsupervised manner: given a video, we want to segment out the objects that are moving independently within it. A video is a sequence of frames $\mathcal{I}_t \in \mathbb{R}^{HW \times 3}$, each of which is an RGB image defined on the lattice $\Omega = \text{vec}(\{1, \dots, H\} \times \{1, \dots, W\}) \in \mathbb{R}^{HW \times 1}$. To segment the objects, we self-supervise a neural network Φ that takes as input each frame $\mathcal{I}_t$ in turn, and outputs a corresponding segmentation mask $\Phi(\mathcal{I}_t) = M_t \in [0, 1]^{HW \times K}$ where K is the number of possible segments we expect to observe in the video. Segmentation matrix entries softly assign each pixel to one of K possible segments.

The challenge is how to supervise the network Φ without labels, utilising only the video itself as training material. The key inductive principle that we propose to use is that physical points that belong to the same object tend to have highly correlated motion, often called *principle of common fate*. When these points are projected to pixels, they result in corresponding highly correlated apparent motions, which we can measure using techniques like optical flow and point tracking. Therefore, we propose to supervise the network Φ from an analysis of apparent motion extracted automatically from the video using off-the-shelf components.

Motion can be measured at two temporal scales. Optical flow extracts instantaneous motion, measuring the 2D velocity of the 3D points found at each pixel in each video frame. Point tracking extracts long-term motion, estimating the 2D location of a certain number of 3D points throughout the video's duration. These two sources of information are complementary. Optical flow is dense, easy to extract, and

easy to model to discover correlations within it; however, by considering different times in isolation, it ignores most of the correlations that exist in the data. Tracks are sparse, more difficult to extract and harder to model, but potentially contain information ignored by optical flow.

Prior works such as [Choudhury et al. 2022] have studied how to model optical flow for segmentation. Here, motivated by a new generation of high-quality point trackers [Karaev et al. 2024; Doersch et al. 2023; Doersch et al. 2022; Harley et al. 2022], we aim at developing the machinery necessary to use track information as well. From this analysis, we construct losses which assess the quality of the predicted mask M_t given the video itself. Next, we introduce two such losses, one for optical flow from prior work, and a new one based on point tracking.

5.3.1 Learning from optical flow

First, we describe the case of optical flow. Because optical flow is instantaneous, we can fix our attention on a specific frame $\mathcal{I}$ and corresponding mask M, dropping for now the time index t. The *optical flow* $F \in \mathbb{R}^{HW\times 2}$ for this image associates a 2-dimensional flow vector to each of the $H \times W$ pixels. Each flow vector can be understood as the velocity of the pixel.

Let $M_k \in \mathbb{R}^{HW\times 1}$ be the binary matrix for segment k, obtained by extracting the k-th column of M. Let $F_k = M_k \odot F$ denote the Hadamard (element-wise) product between the mask and flow vectors, broadcasting the mask along the rows.

Assuming that the object is rigid, the optical flow can be approximated as a linear parametric model of 2D coordinate embeddings (see [Adiv 1985] for an overview). Following [Choudhury et al. 2022], we consider a six-dimensional quadratic embedding kernel $\text{emb}([x,\ y]) = [x,\ x^2,\ y,\ y^2,\ xy,\ 1] \in \mathbb{R}^{1\times 6}$ for pixel coordinates $[x,\ y] \in \Omega$ and associate to each region k a corresponding set of 12 parameters $\theta_k \in \mathbb{R}^{6\times 2}$. Optical flow vectors within a region should be expressible as a linear combination of these six basis functions.

We then consider all pixels embeddings stacked in a single matrix $E_k = M_k \odot \text{emb}(\Omega)$ where the product with the soft mask ensures that the embeddings are "active" only if the corresponding pixels are. The optical flow vectors in the region are then

approximated as

$$F_k \approx \hat{F}_k = E_k \hat{\theta}_k \quad \text{where} \quad \hat{\theta}_k = (E_k^\top E_k)^{-1} E_k^\top F_k, \tag{5.1}$$

where $\hat{\theta}_k$ is obtained via least square. We can use the residual of this approximation as a measure of how well the mask M_k fits the data:

$$\mathcal{L}_f(M|F) = \sum_k \|F_k - \hat{F}_k\|_F^2 = \sum_k \|F_k - E_k \hat{\theta}_k\|_F^2 . \tag{5.2}$$

Intuitively, this considers the correlation of pixel motion in the *spatial* sense: how pixel coordinates determine its motion based on motion parameters θ_k.

5.3.2 Learning from trajectories

Having covered optical flow, we move now to developing an analogous loss for tracking. We write $P \in \mathbb{R}^{2T \times N}$ for the track matrix, with one trajectory per column. With slight abuse of notation, we write $(P)_t \in \mathbb{R}^{2 \times N}$ for indexing rows corresponding to point locations at some time t. To connect pixel-wise masks and sparse points, we use a sampling operation $\pi(\cdot)$, writing $\pi(M_k, (P)_t) = \hat{M}_k \in [0,1]^{N \times 1}$ for mask values at point locations at an appropriate time. Furthermore, we denote by $P_k = P \odot \hat{M}_k$ the masked version of the trajectory matrix, selecting the columns/trajectories that belong to object k with obvious broadcasting of the mask values.

Unlike optical flow, trajectories are too complex to be modelled using a small set of *fixed* basis functions. Instead, we posit that the set of trajectories should be low-rank — all trajectories belonging to the same object should be explained well by a linear combination of some small number of trajectories. We illustrate this intuition in fig. 5.1 using a 2D example.

This assumption results in a factorization of P_k using singular value decomposition (SVD) as $P_k = U_k \Sigma_k V_k^\top$, where $(U_k, \Sigma_k, V_k) = \text{SVD}(P_k)$. As P_k should be low-rank, we can thus form an approximation using truncated SVD, by considering only first r components. We write $\lfloor U_k \rfloor_r$ to denote such truncation. With this, we obtain the loss

$$\mathcal{L}_{\text{rec@}r}(M|P) = \sum_k \left\| P_k - \lfloor U_k \rfloor_r \lfloor \Sigma_k \rfloor_r \lfloor V_k \rfloor_r^\top \right\|_F^2 . \tag{5.3}$$

Since truncated SVD offers optimal decomposition for the error above, lowering

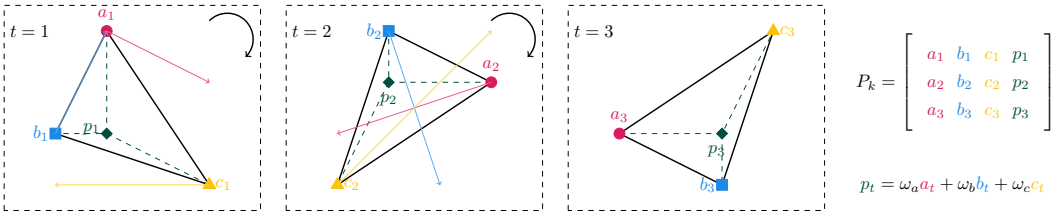


Figure 5.1: **Illustrative 2D example for the low-rank nature of P_k.** A triangle undergoes rigid rotation over three frames. As the rate of rotation is not constant, the flow vectors and point positions are difficult to model. However, the point p is part of the triangle and can be expressed as a combination of the three vertices at an appropriate time. Thus, the last column of P_k is linearly dependent, and P_k is rank deficient. *Any points in the triangle could be included in P_k without increasing its rank.*

this loss amounts to making P_k as close as possible to rank r, i.e., by grouping trajectories into P_k that do not increase its rank, and should come from rigid objects.

As we show in section 5.5.3, we found an alternative formulation of this idea works better. Note the rank r matrix has the r-th and all later singular values as 0. We can optimise singular values higher than r-th to be close to 0 (ignoring U_k and V_k). Thus, for trajectories, we formulate a loss simply as:

$$\mathcal{L}_t(M|P) = \sum_k \sum_{i=r}^{\min(2T,N)} \sigma_i(P_k), \tag{5.4}$$

where $\sigma_i(P_k)$ is the i-th singular value of P_k. We assume $r \ll \min(2T, N)$.

Meaning of decomposition. We show that under certain simplifying assumptions, the decomposition in (5.3) is exact and models time-varying camera motion and object geometry as two terms. We consider a simple case of a rigid body motion observed through a perspective camera. For points on the object, we can consider only the relative motion between the camera and the object and attribute it all to the camera for simplicity.

Given (stacked) camera projection matrices $W_t \in \mathbb{R}^{3T\times 4}$, points $\tilde{X}_k \in \mathbb{R}^{4\times N}$ in homogenous coordinates that remain at constant projective depth $\mathbf{d} \in \mathbb{R}^{N\times 1}$ from the camera over the whole sequence, we note the following equation [Hartley and A. Zisserman 2004]:

$$\tilde{P}_k = W_t \tilde{X}_k \,\mathrm{diag}(\mathbf{d})^{-1}, \tag{5.5}$$

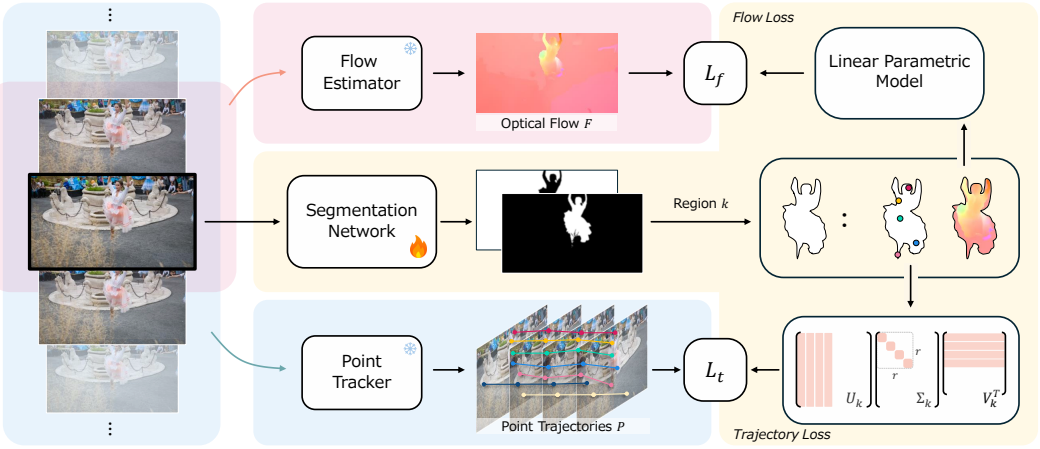


Figure 5.2: **Overview of our approach.** We self-supervise a segmentation network, i.e. without access to mask annotations, using both short-term motion information (optical flow) and long-term motion (point trajectories). We design a loss function that encourages the segmentation network to cluster regions where trajectories form low-rank-r groups, which should align well with objects. Off-the-shelf methods are used to estimate optical flow and point trajectories given a dataset of videos.

where $\tilde{P}_k \in \mathbb{R}^{3T\times N}$ is P_k in homogenous coordinates. Both W_t and $\tilde{X}_k \operatorname{diag}(\mathbf{d})^{-1}$ can be recovered by considering a truncated SVD at rank 4: $W_t = \lfloor U_k \rfloor_4 \lfloor \Sigma_k \rfloor_4$, and $\tilde{X}_k \operatorname{diag}(\mathbf{d})^{-1} = \lfloor V_k \rfloor_4^\top$.

The trajectory matrix factorises into the time-varying camera matrices and object geometry. As the depth is not constant in the real-world setting, this decomposition is approximate and suggests the following alternative loss:

$$\mathcal{L}_{\text{per}} = \sum_k \|\tilde{P}_k - W_t \tilde{X}_k \operatorname{diag}(\mathbf{d})^{-1}\|_F^2, \tag{5.6}$$

where W_t, and $\tilde{X}_k \operatorname{diag}(\mathbf{d})^{-1}$ are obtained via SVD as above.

Choice of r. Setting r correctly is important. Intuitively, it captures the degrees of freedom present in the trajectory data or the number of trajectories that are sufficient to form a basis. From the analysis above, we saw that rank $r = 4$ corresponds to assuming constant depth and perspective camera. However, higher r is needed to tolerate changing depth and tracking errors [Hartley and A. Zisserman 2004; J. P. Costeira and Kanade 1998]. Similarly, not all motion is rigid in real-world videos, which also requires increasing r. We empirically determined $r = 5$ to yield good results.

5.3.3 Training a segmenter using flow and trajectories

The losses above require optical flow F, trajectories P, and masks M_k obtained using a segmentation network $\Phi(\mathcal{I}) = M$. This suggests a simple procedure of training a segmentation network given a dataset of videos, which we summarise in fig. 6.2. We precalculate optical flow for each frame and obtain a set of point trajectories for each video using off-the-shelf pretrained networks. For training, we consider triples of $(\mathcal{I}, F, P)_i$ for each frame i, where for trajectories P, we take trajectories for which the points are visible in the image $\mathcal{I}$. This can be accomplished by making use of visibility predictions in the output of point trackers or calculating trajectories by querying points in each frame. We use bilinear sampling for $\pi(\cdot)$ to obtain mask values at trajectory coordinates.

Temporal smoothing. We include a temporal smoothing loss, which matches mask predictions between two frames offset by Δt using the predicted trajectories:

$$\mathcal{L}_\tau = \|\pi(\Phi(\mathcal{I}_t), (P_t)_t) - \pi(\Phi(\mathcal{I}_{t+\Delta t}), (P_t)_{t+\Delta t})\|_2^2 , \tag{5.7}$$

where $\mathcal{I}_t$ is the t-th frame and P_t are trajectories associated with t-th frame. We write the final loss as: $\mathcal{L} = \lambda_f \mathcal{L}_f + \lambda_t \mathcal{L}_t + \lambda_\tau \mathcal{L}_\tau$, where $\lambda_f, \lambda_t, \lambda_\tau$ balance the contribution of the different loss terms.

Choice of k. Following prior work [Choudhury et al. 2022], we set k, the number of predicted masks, to be higher than the maximum number of objects in the scene to account for potential parallax and non-rigid motion. In the binary segmentation case, we recover two components by considering the average appearance feature of each component and solving for the normalised cut on a graph with k nodes.

5.4 Feasibility study

Our proposed trajectory loss (5.4) enables training a segmentation network using trajectory data. We first show the feasibility of the proposed cost function in a controlled setting, without actually training Φ. To this end, we consider a synthetic scene from the MOVI-F Kubric [Greff et al. 2022] dataset for which we obtain ground-truth trajectories for every point and ground-truth object segmentation

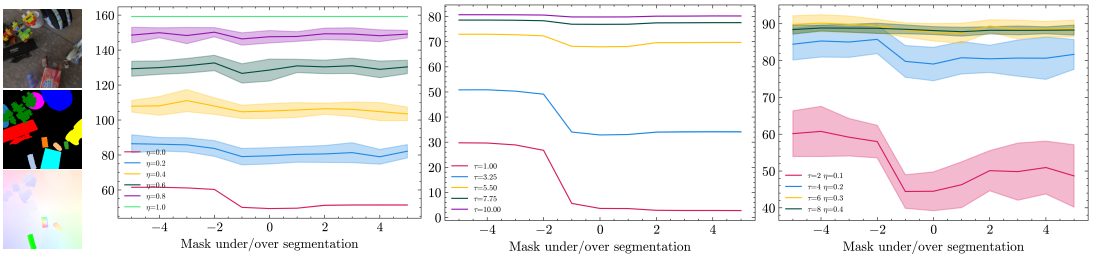


Figure 5.3: **Feasibility analysis of $\mathcal{L}_t$.** Using a synthetic sequence (left), we vary the amount of noise η injected into the mask, the temperature τ of the mask logits and plot the loss value as a function of the mask under/over segmentation. The plots show that the loss is reduced in low-noise, low-entropy settings and penalises both over- and under-segmentation.

masks. We explore the loss landscape of the proposed formulation by corrupting the segmentation masks along several principled axes and studying the effect of such corruptions on the trajectory loss.

First, we consider a random alteration of mask pixels, which we refer to as *mask noise*. We control the amount of mask noise using η such that 0.0 corresponds to no pixels changed and 1.0 corresponds to completely random masks. Along this axis, we test whether our loss favours predictions with lower noise. Second, we consider structural alterations, namely under/over-segmentation. To simulate under-segmentation, we merge object masks with the background at random. To simulate over-segmentation, we randomly split the existing object mask into two parts in the middle along either the x or y-axis. We represent this type of mask corruption using integers. Negative values indicate the number of objects removed, while positive values correspond to new objects generated from existing ones. Such structural corruption investigates whether the loss can correctly identify the number of moving objects. Finally, we consider the "softness" of the predicted masks by transforming masks into logits and increasing the temperature τ in the softmax operation. This tests whether the loss will prefer low-entropy values. We leave further details of the corruption procedure to appendix E.4.

The results of these analyses are shown in Figure 5.3. All three plots show the loss value as a function of structural corruption. The trajectory loss decreases as the noise and temperature of the masks are reduced, as seen in the first two plots. The third plot also shows that such solutions are preferred in combination. Furthermore, we observe that the loss values are lower when the correct number of segments is detected, and this holds even in the presence of noise or when masks

are more uniform. Note, however, that over-segmentation is penalised less than under-segmentation, i.e. missing moving objects leads to a higher value of the loss than, e.g. splitting an object into several components.

5.5 Experiments

In this section, we evaluate our approach for unsupervised motion segmentation and compare it with simple baselines and prior subspace clustering methods. Next, we compare our method with state-of-the-art methods for unsupervised video object segmentation across several datasets in a binary segmentation setting. We finish with ablation experiments of our approach.

Datasets. We consider four primary datasets in this study. We use the synthetic MOVi-F variant of the Kubric [Greff et al. 2022] dataset with ground truth trajectories for comparison with subspace clustering-based approaches. We adopt this setting to eliminate noise in point trajectories as previous methods are sensitive to it. We report the adjusted Rand index (ARI) as the main metric, measuring how close clustering is to the ground truth up to the permutation of cluster identities, where 1 is a perfect match, and 0 means roughly random assignment. We also report FG-ARI, i.e. ARI only on foreground pixels (determined by ground truth masks), which identifies how well different objects are separated.

We also evaluate our approach on real-world datasets: DAVIS 2016 [Perazzi et al. 2016], SegTrackv2 (STv2) [F. Li et al. 2013], and FBMS [Ochs et al. 2014], which are popular benchmarks for video object segmentation. Following standard practice [Charig Yang et al. 2021; Yanchao Yang et al. 2019], foreground objects in STv2 and FBMS are consolidated. We report the Jaccard ($\mathcal{J}$) score, computed using Hungarian matching between predicted and ground truth segmentations.

Implementation. For the experiments on real-world datasets, optical flow is estimated using RAFT [Teed and J. Deng 2020] and point trajectories using CoTracker [Karaev et al. 2024]. Trajectories are computed within a context window $f = 20$ around each frame, with reflection padding around video boundaries, resulting in chunks of $T = 2f + 1 = 41$ frames. To reduce the effect of noisy predictions, we also filter trajectories along the time dimension using an average

filter with a window size of 11. For the experiments on MOVi-F, a small U-Net [Ronneberger et al. 2015] is trained as the segmentation network, starting from random initialisation. For fairness of comparisons on DAVIS, STv2 and FBMS, we use the same architecture as in [Choudhury et al. 2022] — MaskFormer with DINO backbone. We specify further details in appendix E.5.

5.5.1 Comparison to trajectory-based methods

In table 5.1, we compare our low-rank trajectory loss (LRTL) with prior subspace clustering approaches in a per-video optimisation setting. Subspace clustering operates on a similar intuition to our proposed trajectory loss by a grouping of trajectories that should be linearly dependent. We also consider K-means clustering of trajectories as a simple baseline. For fair comparisons, we train our segmentation model optimising *only* the trajectory loss ($\mathcal{L}_t$). We use $k = 25$ components for each video and train for 5000 steps. This is comparable to the computation requirements and steps of other methods. For K-means, SSC [Elhamifar and Vidal 2013] and LRR [G. Liu et al. 2012], we search for an optimal set of hyperparameters and the number of components k, reporting the best results. Our approach shows significantly stronger performance than simple K-Means and subspace clustering approaches.

Table 5.1: Comparison of our LRTL trajectory-based formulation with prior methods.

	MOVi-F	
Method	ARI↑	FG-ARI↑
K-Means	15.26	42.53
SSC[Elhamifar and Vidal 2013]	11.12	39.21
LRR[G. Liu et al. 2012]	7.47	37.36
LRTL (Ours)	46.07	65.76

5.5.2 Unsupervised video object segmentation

We compare to recent methods on the unsupervised video object segmentation task *without first-frame prompting or post-processing*. In this setting, we train a single network on the benchmark datasets for binary video segmentation. We compare with *single-sequence methods* that perform optimisation for each sequence/video individually. Additionally, we benchmark dataset-wide *single-stage end-to-end methods* where training is performed over multiple videos simultaneously, training a network in an end-to-end manner. We also compare with *multi-stage methods* that train and re-train several networks. We report our results on standard benchmarks in table 5.2. While the closest prior work relies on multiple stages of training,

Table 5.2: **Unsupervised video segmentation** on DAVIS, SegTrackv2, and FBMS. Where possible, we report results without widely applicable post-processing (e.g. CRF) or indicate results in grey.

Method	Inf. Input RGB	Inf. Input Motion	Input Resolution	Motion Est. Method	DAVIS $\mathcal{J}\uparrow$	STv2 $\mathcal{J}\uparrow$	FBMS $\mathcal{J}\uparrow$
Single-sequence methods							
FTS[Papazoglou and Ferrari 2013]	✓	✓	–	LDOF[Brox and Malik 2010a]	55.8	47.8	47.7
CUT[Keuper et al. 2015]	✓	✓	–	LDOF[Brox and Malik 2010a]	55.2	54.3	57.2
DS[Ye et al. 2022]	✓	✓	240 × 426	RAFT[Teed and J. Deng 2020]	79.1	72.1	71.8
Ponimatkin et al.[Ponimatkin et al. 2023]	✓	✗	480 × 848	ARFlow[Liang Liu et al. 2020]	80.2	74.9	70.0
OCLR[Xie et al. 2022] (test ft.)	✓	✓	480 × 848	RAFT[Teed and J. Deng 2020]	80.9	72.3	69.8
Single-stage end-to-end methods							
OCLR[Xie et al. 2022]	✗	✓	112 × 224	RAFT[Teed and J. Deng 2020]	72.1	67.6	65.4
DivA[Lao et al. 2023]	✓	✓	128 × 224	RAFT[Teed and J. Deng 2020]	72.4	64.6	60.9
Meunier et al.[Meunier and Bouthemy 2023b]	✗	✓	128 × 224	RAFT[Teed and J. Deng 2020]	73.2	55.0	–
GWM[Choudhury et al. 2022]	✓	✗	128 × 224	RAFT[Teed and J. Deng 2020]	79.5	78.9	78.4
Multi-stage methods							
RCF[Lian et al. 2023]	✓	✗	480 × 848	RAFT[Teed and J. Deng 2020]	80.9	76.7	69.9
LOCATE[S. Singh et al. 2023]	✓	✗	480 × 848	ARFlow[Liang Liu et al. 2020]	80.9	79.9	68.8
LRTL (Ours)	✓	✗	192 × 352	RAFT[Teed and J. Deng 2020] CoTracker[Karaev et al. 2024]	**82.2**	**81.2**	**79.6**

pseudo-labelling, applying CRF, and retraining, our end-to-end trained method shows better performance at lower resolutions. We attribute this to the effectiveness of our approach in incorporating long-term motion information.

In fig. 5.4, we show qualitative results of our approach and compare with RCF [Lian et al. 2023], a state-of-the-art multi-stage approach. Our network trained with both flow and trajectory losses yields segmentations with noticeably better boundaries despite operating at a lower resolution. Notably, our formulation also effectively avoids segmenting shadows and water ripples of the swan, which are difficult to separate based on instantaneous motion alone.

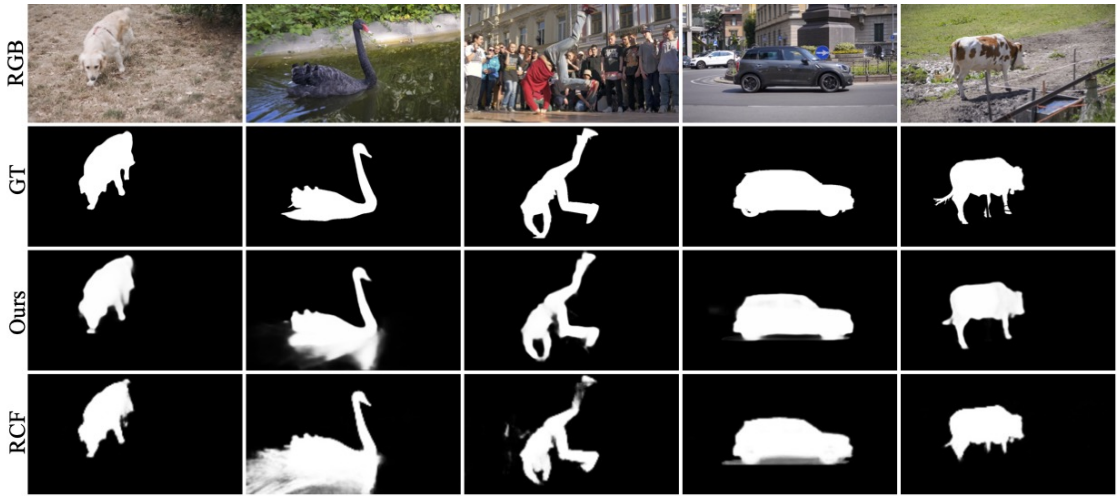


Figure 5.4: Qualitative comparison of our results on DAVIS with RCF which uses higher resolution and multi-stage training. Our method contains slightly better boundaries, does not segment shadows and separates water ripples from the swan.

Table 5.3: **Alternative losses** to our proposal. Other variants do not match the performance of our formulation.

Loss	DAVIS ($\mathcal{J}\uparrow$)
$\mathcal{L}_{\text{rec@3}}$ (5.3)	11.1
$\mathcal{L}_{\text{per}}$ (5.6)	18.2
$\mathcal{L}_{\text{rec@5}}$ (5.3)	14.6
tracks-as-flow	65.3
Ours $\mathcal{L}_t$ (5.4)	71.9

Table 5.4: **Ablation of loss terms.** All loss terms synergise to improve performance.

Loss	DAVIS ($\mathcal{J}\uparrow$)
$\lambda_f \mathcal{L}_f$	78.5
$\lambda_t \mathcal{L}_t$	71.9
$\lambda_t \mathcal{L}_f + \lambda_t \mathcal{L}_t$	81.7
$\lambda_t \mathcal{L}_f + \lambda_t \mathcal{L}_t + \lambda_\tau \mathcal{L}_\tau$	82.2

5.5.3 Ablations

Alternative losses. We have explored several alternative formulations of the trajectory loss in our approach and present the analysis in table 5.3. Losses based on full SVD reconstruction fail to train a network sufficiently. $\mathcal{L}_{\text{per}}$ performs the best out of these, likely as DAVIS contains several scenes with a panning camera tracking a rigid object at an approximately constant distance, which matches the assumptions. Increasing or decreasing the rank of the approximation performs worse. We also consider *track-as-flow* loss, where trajectories P are treated as optical flow by subtracting positions from adjacent times. Then, for T frames, eq. (5.2) can be applied. We find that such a formulation still underperforms in comparison to our trajectory-based formulation (eq. (5.4)).

We believe our formulation provides better results than the above for two possible reasons. First, by minimising higher-than-r singular values, we are not *strictly* enforcing assumptions like rigidity. Second, our loss formulation is more numerically stable as it requires only gradients w.r.t. to the singular values. As we seek to drive them close to zero, the matrices P_k become increasingly ill-conditioned as the training progresses. Additionally, gradients w.r.t. U and $V^\top$ depend on inverse singular values Σ^{-1} [Townsend 2016], which become numerically unstable as they are approaching zero. On the other hand, $\mathrm{d}\,\Sigma = I_N \circ (U^\top \,\mathrm{d}\,P_k\, V)$ does not have this problem.

Influence of losses. In table 5.4, we consider the method with only the flow loss component and only the trajectory loss component. We find that our trajectory-based loss improves flow-only performance. Using only trajectory-based loss shows weaker performance than just optical flow, likely due to using only a sparse set of points and the noise introduced by estimating positions for occluded points.

Ablating temporal smoothing loss slightly lowers performance as well.

Limitations. While we have demonstrated the effectiveness of learning segmentation from long-term motion, there is potential for further improvements in leveraging point trajectories. First, while modern trackers predict reasonable positions for occluded points, naturally, these predictions are less accurate. Thus, a more explicit handling of occlusions and tracking noise would likely help. Second, we currently only use trajectory estimates from nearby frames for training. This means that we sometimes track the same point multiple times, which could be avoided with caching trajectories. While we handle non-rigidity using over-segmentation, extending this principle to video with multiple non-rigid objects is an important feature direction.

5.6 Conclusion

We have introduced a principled method to train an image segmentation network using long-term motion information expressed as point trajectories. Our trajectory loss formulation follows the principle of common fate and aims to group trajectories into low-rank matrices, representing the idea the motion of points belonging to the same object can be roughly explained as a combination of other points. Using synthetic data we have shown that such a loss should prefer low-noise and low-entropy solutions as well as identify the correct number of moving objects. In comparison with other methods, our loss formulation has shown superior performance compared to subspace clustering baselines on synthetic data and achieved state-of-the-art results on unsupervised video object segmentation benchmarks when combined with optical flow-based loss.

Acknowledgements L. K. is supported by is supported by EPSRC AIMS CDT EP/S024050/1. I. L., C. R. and A. V. are supported by ERC-CoG UNION 101001212 and EPSRC VisualAI EP/T028572/1.

Statement of authorship can be found in appendix A. Supplementary material is available in appendix E.

Chapter 6

Diffusion Models for Open-Vocabulary Segmentation

In previous chapters, we explored how motion can be used to find and segment objects in images and videos. This can be thought of as a bottom-up approach where a low-level signal is used to induce appropriate groupings. We now turn to investigating language as a source of information to learn about various objects. Language already provides a rich structure with different terms used to identify and describe objects. This structure can be used to guide the discovery of objects and their appearances in a top-down manner. In this chapter, we explore how *naming* (P4) enables delineating objects in images. We propose a method that adapts a language-conditioned diffusion model to construct an open-vocabulary segmentation model without any additional training or fine-tuning. This shows that text-to-image generative diffusion models already learn necessary information about various objects, which can be used to segment them.

This chapter presents a paper has been published in European Conference on Computer Vision (ECCV), 2024. Additionally, it has been awarded an Oral presentation.

Diffusion Models for Open-Vocabulary Segmentation

Laurynas Karazija, Iro Laina, Andrea Vedaldi, Christian Rupprecht
Visual Geometry Group
University of Oxford
Oxford, UK
{laurynas,iro,vedaldi,chrisr}@robots.ox.ac.uk

Abstract

Open-vocabulary segmentation is the task of segmenting anything that can be named in an image. Recently, large-scale vision-language modelling has led to significant advances in open-vocabulary segmentation, but at the cost of gargantuan and increasing training and annotation efforts. Hence, we ask if it is possible to use *existing* foundation models to synthesise on-demand efficient segmentation algorithms for specific class sets, making them applicable in an open-vocabulary setting without the need to collect further data, annotations or perform training. To that end, we present OVDiff, a novel method that leverages generative text-to-image diffusion models for unsupervised open-vocabulary segmentation. OVDiff synthesises support image sets for arbitrary textual categories, creating for each a set of prototypes representative of both the category and its surrounding context (background). It relies solely on pre-trained components and outputs the synthesised segmenter directly, without training. Our approach shows strong performance on a range of benchmarks, obtaining a lead of more than 5% over prior work on PASCAL VOC.

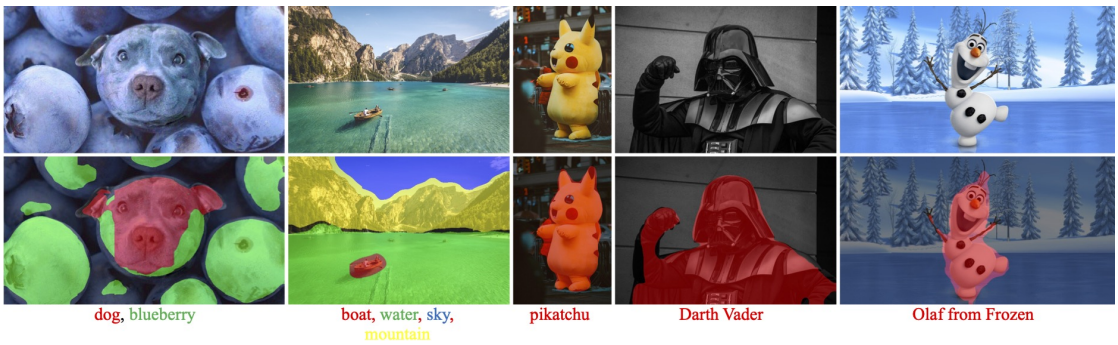


Figure 6.1: OVDiff is an open-vocabulary segmentation method that, given an image and a free-form set of class names, can segment any user-defined classes. It is fully automatic and does not require any further training.

6.1 Introduction

Open-vocabulary semantic segmentation is the task of segmenting images into regions matching several free-form textual categories. As the field of Computer Vision moves towards large-scale general-purpose models, open-vocabulary "foundation" models have similarly emerged. Yet, the development of ones suitable for dense localisation tasks such as semantic segmentation incurs both enormous training costs and requires expensive mask annotations. Instead, we show that the open-vocabulary segmentation task can be effectively tackled starting from a set of frozen foundation models, without requiring additional data or even fine-tuning.

In order to do so, we introduce OVDiff, a method that turns existing foundation models into a "factory" of image segmenters, i.e., using foundation models to synthesise on-demand a segmenter for any new concepts specified in natural language. Thus, OVDiff can be used for open-vocabulary segmentation, where it achieves state-of-the-art results in standard benchmarks. Moreover, once synthesised, the segmenters can be efficiently applied to any number of images and easily extended to new categories.

Specifically, segmenting an image using OVDiff can be done in three steps: *generation*, *representation*, and *matching*. Given a textual prompt, OVDiff uses an off-the-shelf text-to-image generator like StableDiffusion [Rombach et al. 2022] to *generate* a support set of images. In the representation step, we use a feature extractor (that can be the same network as in the generation step) to extract feature prototypes that represent the textual category. Finally, we use simple nearest-

neighbour *matching* scheme to segment the target image using the prototypes computed in the previous step.

This approach differs from prior work that largely approaches the problem in either of two ways. Starting from multi-modal representations (e.g., CLIP [Radford et al. 2021]) to bridge vision and language, the first way relies on labelled data to fine-tune image-level representations for the segmentation task. Hence, in line with the zero-shot setting [Bucher et al. 2019], these methods require costly dense annotations for some known categories while also extending the segmentation to unseen categories by incorporating language.

The second category of prior work [Jiarui Xu et al. 2022; Ren et al. 2023; Jilan Xu et al. 2023; H. Luo et al. 2023; Mukhoti et al. 2023; Cha et al. 2023] observes that large-scale vision-language models such as CLIP have a limited understanding of the positioning of objects within an image and extend these models with additional grouping mechanisms for better localisation using only image-level captions, but no mask supervision. This, however, requires expensive additional contrastive training at scale. Additionally, most methods resort to heuristics to segment the background (i.e., leave some pixels unlabelled), as it often cannot be described as a textual category. The usual approach is to threshold the similarities to all categories. Finding an appropriate threshold, however, can be challenging and may vary depending on the image, often resulting in imprecise object boundaries. Effectively handling the background remains an open issue.

Our three-step approach departs substantially from both of these schemes. We show that large-scale text-to-image generative models, such as StableDiffusion [Rombach et al. 2022], can help bridge the vision-and-language gap without the need for annotations or costly training. Furthermore, diffusion models also produce latent spaces that are semantically meaningful and well-localised. This solves a second problem: multi-modal embeddings are difficult to learn and often suffer from ambiguities and differences in detail between modalities. Instead, our approach can use unimodal features for open-vocabulary segmentation, which offers several advantages. Firstly, as text-to-image generators encode a distribution of possible images, this offers a means to deal with intra-class variation and captures the ambiguity in textual descriptions. Secondly, the generative image models encode not only the visual appearance of objects but also provide contextual priors, which

we use for direct background segmentation.

This work presents a simple framework that achieves state-of-the-art performance across open-vocabulary segmentation benchmarks. It combines several off-the-shelf pre-trained networks into a segmenter "factory" that segments images into arbitrary textual categories in three simple steps. OVDiff requires no additional data, mask supervision, nor fine-tuning. To summarise, we make the following core contributions: (1) We introduce a method to use pre-trained diffusion models for the task of open-vocabulary segmentation, that requires no additional data, mask supervision, or fine-tuning. (2) We propose a principled way to handle backgrounds by forming prototypes from contextual priors built into text-to-image generative models. (3) A set of additional techniques for further improving performance, such as multiple prototypes, category filtering and "stuff" filtering.

6.2 Related work

Zero-shot open-vocabulary segmentation. Open-vocabulary semantic segmentation is a relatively new problem and is typically approached in two ways. The first line of work poses the problem as "zero-shot", i.e., segmenting unseen classes after training on a set of observed classes with dense annotations. Early approaches [Bucher et al. 2019; P. Li et al. 2020; Gu et al. 2020; J. Cheng et al. 2021] explore generative networks to sample features using conditional language embeddings for classes. In [Xian et al. 2019; B. Li et al. 2021] image encoders are trained to output dense features that can be correlated with word2vec [Mikolov et al. 2013] and CLIP [Radford et al. 2021] text embeddings. Follow-up works [Ghiasi et al. 2022; Liang et al. 2023; J. Ding et al. 2022; M. Xu et al. 2022] approach the problem in two steps, predicting class-agnostic masks and aligning the embeddings of masks with language. IFSeg [Yun et al. 2023] generates synthetic feature maps by pasting CLIP text embeddings into a known spatial configuration to use as additional supervision. Different from our approach, all these works rely on mask supervision for a set of known classes.

The second line of work eliminates the need for mask annotations and instead aims to align image regions with language using only image-text pairs. This is largely enabled by recent advancements in large-scale vision-language models [Radford et al.

2021]. Some methods introduce internal grouping mechanisms such as hierarchical grouping [Jiarui Xu et al. 2022; Ren et al. 2023; Wysoczańska et al. 2024], slot-attention [Jilan Xu et al. 2023], or cross-attention to learn cluster centroids [Q. Liu et al. 2022; H. Luo et al. 2023]. Assignment to language queries is performed at group level. Another line of work [C. Zhou et al. 2022; Mukhoti et al. 2023; Cha et al. 2023; Ranasinghe et al. 2023] aims to learn dense features that are better localised when correlated with language embeddings at pixel level. With the exception of [Ranasinghe et al. 2023; C. Zhou et al. 2022; Wysoczańska et al. 2024], thresholding is often required to determine the background during inference. Alternatively, a curated list of background prompts can be used [Ranasinghe et al. 2023].

Our method falls into the second category. However, in contrast to prior work, we leverage a generative model to translate language queries to pre-trained image feature extractors without further training. We also segment the background directly, without relying on thresholding or curated list of background prompts. A closely related approach to ours is ReCO [Shin et al. 2022b], where CLIP is used for image retrieval compiling a set of exemplar images from ImageNet for a given language query, which is then used for co-segmentation. In our method, the shortcoming of an image database is addressed by synthesising data on-demand. Furthermore, instead of co-segmentation, we leverage the cross-attention of the generator to extract objects. Instead of similarity of support images, we use diverse samples and both foreground and contextual backgrounds. Follow up works [Barsellotti et al. 2024a; Barsellotti et al. 2024b] to OVDiff exchange contextual prior for backgrounds with compiling a database of prototypes.

Diffusion models. Diffusion models [Sohl-Dickstein et al. 2015; Ho et al. 2020; Song et al. 2021] are a class of generative methods that have seen tremendous success in text-to-image systems such as DALL-E [Ramesh et al. 2022], Imagen [Saharia et al. 2022], and Stable Diffusion [Rombach et al. 2022], trained on Internet-scale data such as LAION-5B [Schuhmann et al. 2022]. The step-wise generative process and the language conditioning make pre-trained diffusion models attractive also for discriminative tasks. They have been recently used in few-shot classification [Renrui Zhang et al. 2023], few-shot segmentation [Baranchuk et al. 2022] and panoptic segmentation [Jiarui Xu et al. 2023], and to generate pairs of

images and segmentation masks [Z. Li et al. 2023]. However, these methods rely on dense manual annotations to associate diffusion features with the desired output.

Annotation-free discriminative approaches such as [A. C. Li et al. 2023; K. Clark and Jaini 2024; Udandarao et al. 2023] use pre-trained diffusion models as zero-shot classifiers. DiffuMask [W. Wu et al. 2023] uses prompt engineering to synthesise a dataset of "known" and "unseen" categories and trains a closed-set segmenter with masks obtained from the cross-attention maps of the diffusion model. DiffusionSeg [C. Ma et al. 2023] uses DDIM inversion [Song et al. 2021] to obtain feature maps and attention masks of object-centric images to perform unsupervised object discovery, but relies on ImageNet labels and is not open-vocabulary. Our approach also leverages the rich semantic information present in diffusion models for segmentation; unlike these methods, however, it is open-set and does not require further training.

Unsupervised segmentation. Our work is also related to unsupervised segmentation approaches. While early works relied on hand-crafted priors [M.-M. Cheng et al. 2015; Y. Wei et al. 2012; J. Zhang et al. 2018; Y. Zeng et al. 2019; Nguyen et al. 2019] later approaches leverage feature extractors such as DINO [Caron et al. 2021] and perform further analysis of these methods [Y. Wang et al. 2022; Melas-Kyriazi et al. 2022a; Siméoni et al. 2021; Siméoni et al. 2023; Hamilton et al. 2022; Shin et al. 2022a; Xudong Wang et al. 2023a; Xinlong Wang et al. 2022]. Some approaches make use of generative methods, usually GANs, to separate images in foreground and background layers [Bielski and Favaro 2019; M. Chen et al. 2019; Benny and Wolf 2020; Bielski and Favaro 2022] or analyse latent structure to induce known foreground-background changes [Voynov et al. 2021; Melas-Kyriazi et al. 2022b] to synthesise a training dataset with labels. Some works explore interaction with different modalities such as optical flow [Choudhury et al. 2022; Karazija et al. 2022] or depth [Bowen et al. 2022]. Largely focused on unsupervised saliency prediction, these methods are class-agnostic and do not incorporate language.

6.3 Method

We present OVDiff, a method for open-vocabulary segmentation, i.e., semantic segmentation of any category described in natural language. We achieve this

goal in three steps: (1) we leverage text-to-image generative models to *generate* a set of images representative of the described category, (2) use these to ground *representations* from off-the-shelf pretrained feature extractors, and (3) *match* these against input image features to perform segmentation.

6.3.1 OVDiff: Diffusion-based open-vocabulary segmentation

Our goal is to devise an algorithm which, given a new vocabulary of categories $c_i \in \mathcal{C}$ formulated as natural language queries, can segment any image against it. Let $I \in \mathbb{R}^{H \times W \times 3}$ be an image to be segmented. Let $\Phi_v : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^{H' W' \times D}$ be an off-the-shelf visual feature extractor and $\Phi_t : \mathbb{R}^{d_t} \to \mathbb{R}^D$ a text encoder. Assuming that image and text encoders are aligned, one can achieve segmentation by simply computing a similarity function, for example, the cosine similarity $s(\Phi_v(I), \Phi_t(c_i))$, with $s(x, y) = \frac{x^T y}{\|x\| \|y\|}$, between the encoded image $\Phi_v(I)$ and an encoding of a class label c_i. To meaningfully compare different modalities, image and text features must lie in a shared representation space, which is typically learned by jointly training Φ_v and Φ_t using image-text or image-label pairs [Radford et al. 2021].

We propose two modifications to this approach. First, we observe that it is better to compare representations of the *same* modality than across vision and language modalities. We thus replace $\Phi_t(c_i)$ with a D-dimensional *visual* representation $\bar{P}$ of class c_i, which we refer to as a *prototype*. In this case, the same feature extractor can be used for both prototypes and target images; thus, their comparison becomes straightforward and does not necessitate further training. Second, we propose utilising *multiple* prototypes per category instead of a single class embedding. This enables us to accommodate intra-class variations in appearance, and, as we explain later, it also allows us to exploit contextual priors, which in turn help to segment the background.

Our approach, thus, proceeds in three steps: (1) a set of support images is sampled based on vocabulary $\mathcal{C}$, (2) a set of prototypes $\mathcal{P}$ is calculated, and (3) a set of images $\{I_1, I_2 \dots\}$ is segmented against these prototypes. We observe that in practical applications, whole image collections are processed using the same vocabulary, as altering the set of target classes for individual images in an informed

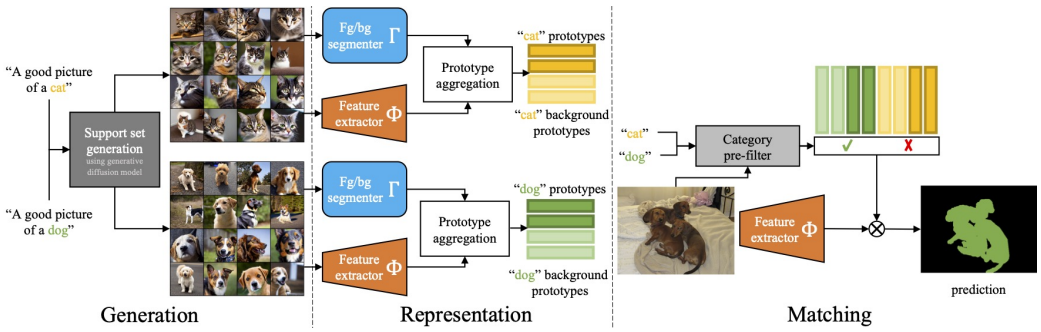


Figure 6.2: OVDiff overview. Prototype sampling: text queries are used to sample a set of support images which are further processed by a feature extractor and a segmenter forming positive and negative (background) prototypes. Segmentation: image features are compared against prototypes.The CLIP filter removes irrelevant prototypes based on global image contents.

way would already require some knowledge of their contents. Steps (1) and (2) are, thus, performed very infrequently, and their cost is heavily amortised. Next, we detail each step.

6.3.2 Support set generation

To construct a set of prototypes, the first step of our approach is to sample a support set of images representative of each category c_i. This can be accomplished by leveraging pretrained text-conditional generative models. Sampling images from a generative model, as opposed to a curated dataset of real images, aligns well with the goals of open-vocabulary segmentation as it enables the construction of prototypes for *any* user-specified category or description, even those for which a manually labelled set may not be readily available (e.g., c_i = "`donut with chocolate glaze`").

Specifically, for each query c_i, we define a prompt "`A good picture of a` $\langle c_i \rangle$" and generate a small batch of N support images $\mathcal{S} = \{S_1, S_2, \ldots, S_N \mid S_n \in \mathbb{R}^{hw \times 3}\}$ of height h and width w using Stable Diffusion [Rombach et al. 2022].

6.3.3 Representing categories

Naïvely, prototypes $\bar{P}_{c_i}$ could be constructed by averaging all features across all images for class c_i. This is unlikely to result in good prototypes because not all pixels in the sampled images correspond to the class specified by c_i. Instead, we propose to extract the class prototypes as follows.

Class prototypes. Our approach generates two sets of prototypes, positive and negative, for each class. Positive prototypes are extracted from image regions that are associated with $\langle c_i \rangle$, while negative prototypes represent "background" regions. Thus, to obtain prototypes, the first step is segmenting the sampled images into foreground and background. To identify regions most associated with c_i, we use the fact that the layout of a generated image is largely dependent on the cross-attention maps of the diffusion model [Hertz et al. 2023], i.e., pixels attend more strongly to words that describe them. For a given word or description (in our case c_i), one can generate a set of attribution maps $\mathcal{A} = \{A_1, A_2, \ldots, A_N \mid A_n \in \mathbb{R}^{hw}\}$, corresponding to the support set $\mathcal{S}$, by summing the cross-attention maps across all layers, heads, and denoising steps of the network [Tang et al. 2023].

Yet, thresholding these attribution maps may not be optimal for segmenting foreground/background, as they are often coarse or incomplete, and sometimes only parts of objects receive high activation. To improve segmentation quality, we propose to optionally leverage an unsupervised instance segmentation method Γ. Unsupervised segmenters are not vocabulary-aware and may produce multiple binary object proposals. We denote these as $\mathcal{M}_n = \{M_{nr} \mid M_{nr} \in \{0,1\}^{hw}\}$, where n indexes the support images and r indexes the object masks (including a mask for the background). We thus construct a promptable extension of Γ segmenter to select appropriate proposals for foreground and background: for each image, we select from $\mathcal{M}_n$ the mask with the highest (lowest) average attribution as the foreground (background):

$$M_n^{\text{fg}} = \arg\max_{M \in \mathcal{M}_n} \frac{M^\top A_n}{M^\top M}, \quad M_n^{\text{bg}} = \arg\min_{M \in \mathcal{M}_n} \frac{M^\top A_n}{M^\top M}. \tag{6.1}$$

Prototype aggregation. We can compute prototypes P_n^{g} for foreground and background regions ($\text{g} \in \{\text{fg}, \text{bg}\}$) as

$$P_n^{\text{g}} = \frac{(\hat{M}_n^{\text{g}})^\top \Phi_v(S_n)}{m_n^{\text{g}}} \in \mathbb{R}^D, \tag{6.2}$$

where $\hat{M}_n^{\text{g}}$ denotes a resized version of M_n^{g} that matches the spatial dimensions of $\Phi_v(S_n)$, and $m_n^{\text{g}} = (\hat{M}_n^{\text{g}})^\top \hat{M}_n^{\text{g}}$ counts the number of pixels within each mask. In other words, prototypes are obtained by means of an off-the-shelf pretrained feature extractor and computed as the average feature within each mask.

We refer to these as *instance* prototypes because they are computed from each image individually, and each image in the support set can be viewed as an instance of class c_i.

In addition to instance prototypes, we found it helpful to also compute *class-level* prototypes $\bar{P}^{\mathrm{g}}$ by averaging the instance prototypes weighted by their mask sizes as $\bar{P}^{\mathrm{g}} = \sum_{n=1}^{N} m_n^{\mathrm{g}} P_n^{\mathrm{g}} / \sum_{n=1}^{N} m_n^{\mathrm{g}}$.

Finally, we propose to augment the set of class and instance prototypes using K-Means clustering of the masked features to obtain *part-level* prototypes. We perform spatial clustering separately on foreground and background regions and take each cluster centroid as a prototype P_k^{g} with $1 \leq k \leq K$. The intuition behind this is to enable segmentation at the level of parts, support greater intra-class variability, and a wider range of feature extractors that might not be scale invariant.

We consider the union of all these feature prototypes:

$$\mathcal{P}^{\mathrm{g}} = \bar{P}^{\mathrm{g}} \cup \{P_n^{\mathrm{g}} \mid 1 \leq n \leq N\} \cup \{P_k^{\mathrm{g}} \mid 1 \leq k \leq K\} \tag{6.3}$$

for $\mathrm{g} \in \{\mathrm{fg}, \mathrm{bg}\}$, and associate them with a single category.

We note that this process is repeated for each $c_i \in \mathcal{C}$ and we hereby refer to $\mathcal{P}^{\mathrm{fg}}$ (and $\mathcal{P}^{\mathrm{bg}}$) as $\mathcal{P}_{c_i}^{\mathrm{fg}}$ ($\mathcal{P}_{c_i}^{\mathrm{bg}}$), i.e., as the foreground (background) prototypes of class c_i.

Since $\mathcal{P}_{c_i}^{\mathrm{fg}}$ ($\mathcal{P}_{c_i}^{\mathrm{bg}}$) depend only on class c_i, they can be precomputed, and the set of classes can be dynamically expanded without the need to adapt existing prototypes.

6.3.4 Segmentation via prototype matching

To perform segmentation of any target image I given a vocabulary $\mathcal{C}$, we first extract image features using the same visual encoder Φ_v used for the prototypes. The vocabulary is expanded with an additional background class $\hat{\mathcal{C}} = \{c_{\mathrm{bg}}\} \cup \mathcal{C}$, for which the positive (*foreground*) prototype is the union of all *background* prototypes in the vocabulary: $\mathcal{P}_{c_{\mathrm{bg}}}^{\mathrm{fg}} = \bigcup_{c_i \in \mathcal{C}} \mathcal{P}_{c_i}^{\mathrm{bg}}$. Then, a segmentation map can simply be obtained by matching dense image features to prototypes using cosine similarity. A class with the highest similarity in its prototype set is chosen:

$$M = \arg\max_{c \in \hat{\mathcal{C}}} \max_{P \in \mathcal{P}_c^{\mathrm{fg}}} s(\Phi_v(I), P). \tag{6.4}$$

Category pre-filtering. To limit the impact of spurious correlations that might exist in the feature space of the visual encoder, we introduce a pre-filtering process for the target vocabulary given image I. Specifically, we leverage CLIP [Radford et al. 2021] as a strong open-vocabulary classifier but propose to apply it in a multi-label fashion to constrain the segmentation to the subset of categories $\mathcal{C}' \subseteq \mathcal{C}$ that appear in the target image. First, we encode the target image and each category using CLIP. Any categories that do not score higher than $1/|\mathcal{C}|$ are removed from consideration, that is we keep the subset $\{P^{\mathrm{g}}_{c'} \mid c' \in \mathcal{C}'\}$, $\mathrm{g} \in \{\mathrm{fg}, \mathrm{bg}\}$. If more than η categories are present, then the top-η are selected. We then form "multi-label" prompts as "$\langle c_a \rangle$ `and` $\langle c_b \rangle$ `and` `...`" where the categories are selected among the top scoring ones taking into account all 2^η combinations. The best-scoring multi-label prompt determines the final list of categories to be used in Equation (6.4).

"Stuff" filtering. Occasionally, c_i might not describe a countable object category but an identifiable region in the image, e.g., `sky`, often referred to as a "stuff" class. "Stuff" classes warrant additional consideration as they might appear as background in images of other categories, e.g., `boat` images might often contain regions of `water` and `sky`. As a result, the process outlined above might sample background prototypes for one class that coincide with the foreground prototypes of another. To mitigate this issue, we introduce an additional filtering step to detect and reject such prototypes, when the full vocabulary, i.e., the set of classes under consideration, is known. First, we only consider foreground prototypes for "stuff" classes. Additionally, any negative prototypes of "thing" classes with high cosine similarity with any of the "stuff" class prototypes are simply removed. In our experiments, we use ChatGPT [OpenAI 2023] to automatically categorise a set of classes as "thing" or "stuff".

Table 6.1: Open-vocabulary segmentation. Comparison of our approach, OVDiff, to the state of the art (under the mIoU metric). Our results are an average of 5 seeds $\pm\sigma$. *results from [Cha et al. 2023].

Method	Support Set	Further Training	VOC	Context	Object
ReCo* [Shin et al. 2022b]	Real	✗	25.1	19.9	15.7
ViL-Seg [Q. Liu et al. 2022]	✗	✓	37.3	18.9	-
MaskCLIP* [C. Zhou et al. 2022]	✗	✗	38.8	23.6	20.6
TCL [Cha et al. 2023]	✗	✓	51.2	24.3	30.4
CLIPpy [Ranasinghe et al. 2023]	✗	✓	52.2	-	<u>32.0</u>
GroupViT [Jiarui Xu et al. 2022]	✗	✓	52.3	22.4	-
ViewCo [Ren et al. 2023]	✗	✓	52.4	23.0	23.5
SegCLIP [H. Luo et al. 2023]	✗	✓	52.6	<u>24.7</u>	26.5
OVSegmentor [Jilan Xu et al. 2023]	✗	✓	53.8	20.4	25.1
CLIP-DIY [Wysoczańska et al. 2024]	✗	✗	<u>59.9</u>	–	31.0
OVDiff (-CutLER)	Synth.	✗	62.8	28.6	34.9
OVDiff	Synth.	✗	**66.3 ± 0.2**	**29.7 ± 0.3**	**34.6 ± 0.3**
TCL [Cha et al. 2023] (+PAMR)	✗	✓	<u>55.0</u>	<u>30.4</u>	<u>31.6</u>
OVDiff (+PAMR)	Synth.	✗	**68.4 ± 0.2**	**31.2 ± 0.4**	**36.2 ± 0.4**

6.4 Experiments

We evaluate OVDiff on the open-vocabulary semantic segmentation task. First, we consider different feature extractors and investigate how they can be grounded by leveraging our approach. We then turn to comparisons of our method with prior work. We ablate the components of OVDiff, visualize the prototypes, and conclude with a qualitative comparison with prior works on in-the-wild images. In appendix F.1, we provide additional experiments concerning image generators, prompting strategies, segmentation methods, and features used.

Datasets and implementation details. As the approach does not require further training of components, we only consider data for evaluation. Following prior work [Jiarui Xu et al. 2022], to assess the segmentation performance, we report mean Intersection-over-Union (mIoU) on validation splits of PASCAL VOC (VOC) [M. Everingham et al. 2012], PASCAL Context (Context) [Mottaghi et al. 2014] and COCO-Object (Object) [Caesar et al. 2018] datasets, with 20, 59, and 80 foreground classes, respectively. These datasets include a background class to reflect a realistic setting of non-exhaustive vocabularies. Context also contains both "things" and "stuff" classes. We also evaluate without background on VOC, Context, ADE20K [B. Zhou et al. 2017], COCO-Stuff [Caesar et al. 2018] and Cityscapes [Cordts et al. 2016], with 20, 59, 150, 171, and 19 classes, respectively,

but do not consider this a realistic setting as it relies on knowing which pixels cannot be described by a set of categories. Similar to [Cha et al. 2023; Jiarui Xu et al. 2022; Jilan Xu et al. 2023], we employ a sliding window approach. We use two scales to aid with the limited resolution of off-the-shelf feature extractors with square window sizes of 448 and 336 and a stride of 224 pixels. We set the size of the support set to $N = 32$. For the diffusion model, we use Stable Diffusion v1.5; for unsupervised segmenter Γ, we employ CutLER [Xudong Wang et al. 2023a].

6.4.1 Grounding feature extractors

Our method can be combined with *any* pretrained visual feature extractor for constructing prototypes and extracting image features. To verify this quantitatively, we experiment with various self-supervised ViT feature extractors (table 6.2): DINO [Caron et al. 2021], MAE [K. He et al. 2022], and CLIP [Radford et al. 2021]. We also use SD as a feature extractor.

We find that SD performs the best, though CLIP and DINO also show strong performance based on our experiments on VOC. MAE shows the weakest performance, which may be attributed to its lack of semanticity [K. He et al. 2022]; yet it is still competitive with the majority of purposefully trained networks when employed as part of our approach. We find that taking *keys* of the second to last layer in CLIP yields better results than using patch tokens (CLIP token). As feature extractors have different training objectives, we hypothesise that their feature spaces might be complementary. Thus, we also consider an ensemble approach. In this case, the cosine distances formed between features of different extractors and respective prototypes are averaged. The combination of SD, DINO, and CLIP performs the best. We adopt this formulation for the main set of experiments.

6.4.2 Comparison to existing methods

In table 6.1, we compare our method with prior work that does not rely on manual mask annotation on three datasets: VOC, Context, Object. We include a brief overview of the methods in the supplement. We find that our method compares favourably, outperforming other methods in all settings. In particular, results on VOC show the largest margin, with more than 5% improvement over prior work.

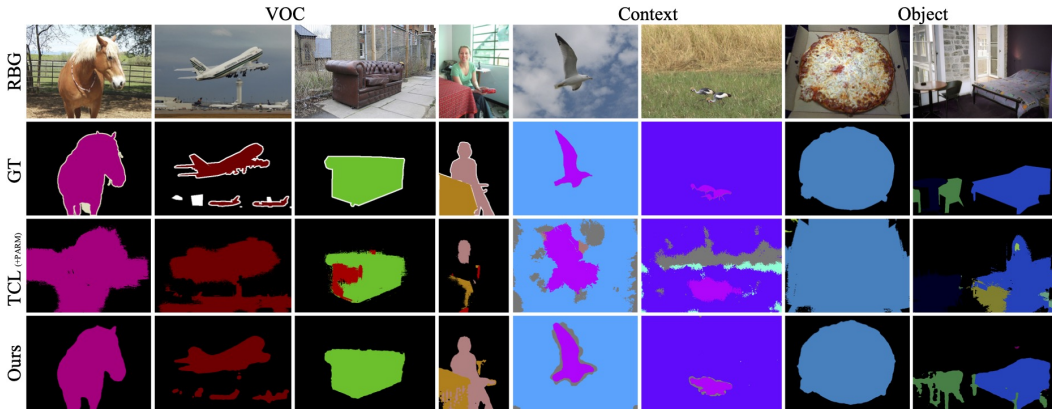


Figure 6.3: Qualitative results. OVDiff in comparison to TCL (+ PAMR). OVDiff provides more accurate segmentations across a range objects and stuff classes with well defined object boundaries that separate from the background well.

We also consider a version of our method, OVDiff (-CutLER), that does not rely on an additional unsupervised segmenter Γ. Instead, the attention masks are thresholded. We observe that such a version of OVDiff has strong performance, outperforming prior work as well. CutLER is helpful, but not a critical component, and OVDiff performs strongly without it.

In the same table, we also combine our method with PAMR [Araslanov and Roth 2020], the post-processing approach employed by TCL. We find that it improves results for our method, though improvements are less drastic since our method already yields better segmentation and boundaries.

Qualitative results are shown in fig. 6.3. This figure highlights a key benefit of our approach: the ability to exploit contextual priors through the use of background prototypes, which in turn allows for the direct assignment of pixels to a background class. This improves segmentation quality because it makes it easier to differentiate objects from the background and to delineate their boundaries. In comparison, TCL predictions are very coarse and contain more noise.

6.4.3 Ablations

Next, we ablate the components of OVDiff on VOC and Context datasets. For these experiments, only SD is employed as a feature extractor. We remove individual components and measure the change in segmentation performance, summarising the results in table 6.3. Our first observation is that background prototypes have a major impact on performance. When removing them from consideration, we

Table 6.2: Performance of OVDiff based on different feature extractors.

Feature Extractor	**VOC**
MAE	54.9
DINO	59.1
CLIP (tokens)	51.4
CLIP (keys)	61.8
SD	64.4
SD+CLIP+DINO	66.4

Table 6.3: Ablation of different components. Each component is removed in isolation, measuring the drop (Δ) in mIoU on VOC and Context datasets. Using SD features.

Configuration	**VOC**	**Δ**	**Context**	**Δ**
Full	64.4		29.4	
w/o bg prototypes	53.2	-11.2	28.9	-0.5
w/o category filter	54.4	-10.0	25.2	-4.2
w/o "stuff" filter	n/a		26.9	-2.5
w/o CutLER	60.4	-4.0	27.6	-1.8
w/o sliding window	62.2	-2.2	28.6	-0.8
only average $\bar{P}$	62.5	-1.9	28.4	-1.0

instead threshold the similarity scores of the images with the foreground prototypes (set to 0.72, determined via grid search); in this case, the performance drops significantly, which again highlights the importance of leveraging contextual priors. On Context, the impact is less significant, likely due to the fact that the dataset contains "stuff" categories. Removing the *instance-* and *part-level* prototypes also negatively affects performance. Additionally, removing the category pre-filtering has a major impact. We hypothesize that this introduces spurious correlations between prototypes of different classes. On Context, "stuff" filtering is also important.

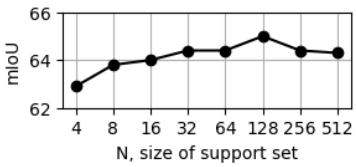


Figure 6.4: PascalVOC results with increasing support size N.

We again consider the importance of using an unsupervised segmenter, CutLER, for prototype mask extractions, using thresholding instead. We find this slightly reduces performance in this setting as well. Overall, background prototypes and pre-filtering contribute the most.

Finally, we measure the effect of varying the size of the support set N in fig. 6.4. We find that OVDiff already shows strong performance even at a low number of samples for each query. With increasing the number of samples, the performance improves, saturating at around $N = 32$. which we use in our main experiments.

Table 6.4: Comparison with methods when background is excluded (decided by ground truth). OVDiff shows comparable performance to prior works despite only relying on pretrained feature extractors. * result from [Cha et al. 2023].

Method	VOC-20	Context-59	ADE	Stuff	Cityscapes
CLIPpy	–	–	13.5	–	–
OVSegmentor	–	–	5.6	–	–
GroupViT*	79.7	23.4	9.2	15.3	11.1
MaskCLIP*	74.9	26.4	9.8	16.4	12.6
ReCo*	57.5	22.3	11.2	14.8	21.1
TCL	77.5	30.3	**14.9**	19.6	23.1
OVDiff	**80.9**	**32.9**	14.1	**20.3**	**23.4**

6.4.4 Evaluation without background

One of the notable advantages of our approach is the ability to represent background regions via (negative) prototypes, leading to improved segmentation performance. Nevertheless, we hereby also evaluate our method under a different evaluation protocol adopted in prior work, which excludes the *background* class from the evaluation. We note that prior work often requires additional considerations to handle background, such as thresholding. In this setting, however, the background class is *not* predicted, and the set of categories, thus, must be exhaustive. As in practice, this is not the case, and datasets contain unlabelled pixels (or simply a background label), such image areas are removed from consideration. Consequently, less emphasis is placed on object boundaries in this setting. As in this setting the background prediction is invalid, we do not consider negative prototypes. This setting tests the ability of various methods to discriminate between different classes, which for OVDiff is inherent to the choice of feature extractors. Despite this, our method shows competitive performance accross wide range of benchmarks table 6.4.

6.4.5 Explaining segmentations

We inspect how our method segments certain regions by considering which prototype from $\mathcal{P}_c^{\text{fg}}$ was used to assign a class c to a pixel. Prototypes map to regions in the support set from where they were aggregated, e.g., instances prototypes are associated with foreground masks M_n^{fg} and part prototypes with centroids/clusters. By following these mappings, a set of support image regions can be retrieved for each segmentation decision, providing a degree of explainability. fig. 6.5 illustrates this for examples of `dog`, `cat`, and `bird` classes. For visualisation purposes, selected

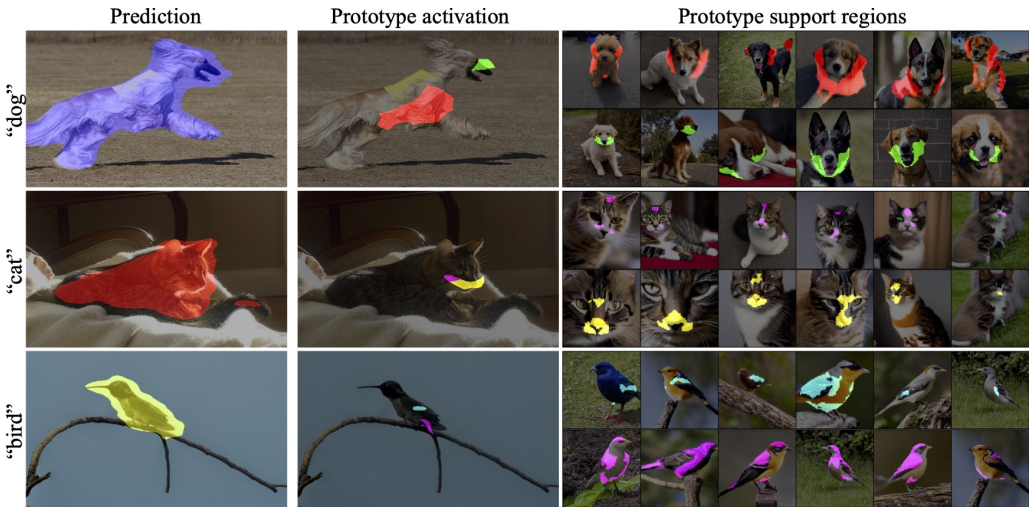


Figure 6.5: Analysis of the segmentation output by linking regions to samples in the support set. Left: our results for different classes. Middle: select color-coded regions "activated" by different prototypes for the class. Right: regions in the support set images corresponding to these (part-level) prototypes.

prototypes and corresponding regions are shown. On the left, we show the full segmentation result of each image. In the middle, we select regions that correlate best with certain class prototypes. On the right, we retrieve images from the support set and highlight where each prototype emerged. We find that meaningful part segmentation merges due to clustering the support image features, and similar regions are segmented by corresponding prototypes. However, sometimes region covered in the input image will not fully align with the whole prototype (e.g. `cat`'s face around the eyes or lower belly/tail of `bird`). Each segmentation is explained by precise regions in a small support set.

6.4.6 In-the-wild

In fig. 6.6, we investigate OVDiff on chal lenging in-the-wild images with simple and complex backgrounds. We compare with TCL+PAMR. In the first three images, both methods correctly detect the objects identified by the queries. OVDiff has small false positive "corgi" patches. TCL however misses large parts of the objects, such as most of the person, and parts of animal bodies. The distinction between the house and the bridge in the second image is also better with OVDiff. We also note that our segmentations sometimes have halos around objects. This is caused by upscaling the low-resolution feature extractor (SD in this case). The last two images contain challenging scenarios where both approaches struggle. The fourth image only contains similar objects of the same type. Both methods incorrectly

Figure 6.6: Qualitative comparison on challenging in-the-wild images with TCL, which struggles with object boundaries, missing parts of objects, or including surroundings. Our method has more appropriate boundaries and makes fever errors overall, but does produce a small halo effect around objects due to the upscaling of feature extractors.

identify plain donuts as either of the specified queries. OVDiff however correctly identifies chocolate donuts with varied sprinkles and separates all donuts from the background. In the final picture, the query "red car" is added, although no such object is present. The extra query causes TCL to incorrectly identify parts of the red bus as a car. Both methods incorrectly segment the gray car in the distance. However, overall, our method is more robust and delineates objects better despite the lack of specialized training or post-processing.

6.5 Conclusion

We introduce OVDiff, an open-vocabulary segmentation method that operates in two stages. First, given queries, support images are sampled and their features are extracted to create class prototypes. These prototypes are then compared to features from an inference image. This approach offers multiple advantages: diverse prototypes accommodating various visual appearances and negative prototypes for background localisation. OVDiff outperforms prior work on benchmarks, exhibiting fewer errors, effectively separating objects from background, and providing explainability through segmentation mapping to support set regions.

Acknowledgements

Laurynas Karazija is supported by is supported by AIMS CDT EP/S024050/1. Iro Laina, Andrea Vedaldi, and Christian Rupprecht are supported by ERC-CoG UNION 101001212 and VisualAI EP/T028572/1.

Statement of authorship can be found in appendix A. Supplementary material is available in appendix F.

Chapter 7

Discussion

7.1 Summary and Impact

This section summarises the contributions of the papers and discusses their impact on the broader field. We then present directions for future work that improve and build upon the ideas presented in this thesis.

7.1.1 Compositional Reconstruction

In chapter 2, we presented a paper, "CLEVRTEX: A Texture-Rich Benchmark for Unsupervised Multi-Object Segmentation". We introduced a series of new benchmarks for unsupervised multi-object segmentation. We also presented a detailed empirical analysis of the effectiveness of various approaches that use *compositional* reconstruction (P2) as a principled mechanism for segmenting multiple objects. Our findings show that various models struggled as the complexity of the scenes increased and failed to achieve satisfactory results when the object and background appearance were no longer uniform.

CLEVRTEX highlighted the limitations of current approaches and challenged the community to consider scaling the compositional reconstruction methods to visually complex scenes. The introduced datasets have become a popular benchmark for unsupervised multi-object segmentation. Various follow-up works proposed modifications to the popular slot attention [Locatello et al. 2020] framework to scale models to more complex settings. These can broadly be grouped into three categories.

The first category explored guiding the attention in the most popular slot-attention [Locatello et al. 2020] mechanism to focus on the objects, thus improving the segmentation performance. Ideas based on promoting cycle consistency [J. Kim et al. 2023; Z. Wang et al. 2023] within pixel-to-object, object-to-part relations have been proposed. In [Biza et al. 2023], invariance to various changes in the object position and rotation for the attention mechanism have been analysed, showing improvements when translation invariance is considered.

The second category of methods looked at the reconstruction quality of the networks and the tendency of compositional models to overfit the scene appearance. This research scaled the decoder network, which reconstructs the input based on slots, to be more powerful. The intuition is that given an expressive decoder, slots only have to encode conditioning signals rather than a detailed encoding of object appearances. Building on the recent success of image generative models, autoregressive [G. Singh et al. 2022a] and diffusion [Jiang et al. 2023] decoders have been shown to improve performance in complex scenes.

The third category of works considered the tendency of compositional models to prefer uniform appearance regions, highlighted in CLEVRTEX. Several proposals turned such a feature of the popular slot-attention [Locatello et al. 2020] mechanism into a strength. These methods replaced the reconstruction target with a modality where the representation does not vary much within the object. For example, in optical flow, the flow vectors are similar within the object but are often very different from the background. [Kipf et al. 2022; Elsayed et al. 2022] explored using optical flow as a reconstruction target. However, as we showed in chapter 4, a more direct analysis incorporating motion is more effective. Alternatively, self-supervised representations are extremely semantic, particularly DINO [Caron et al. 2021]. Principle components of DINO features vary very little within an object. [Seitzer et al. 2023] used this feature space as a reconstruction target, scaling to real-world datasets.

While Seitzer et al. [Seitzer et al. 2023] scaled Slot Attention [Locatello et al. 2020] to be able to learn from real-world data, the approach still remains limited in comparison to alternative approaches using self-supervised features and self-labelling [Xudong Wang et al. 2023a]. Slot Attention-based architectures remain computationally expensive and more difficult to train compared to both specialised

segmentation models and general vision transformers. This presents a scaling challenge. However, obtaining object-centric representations [Greff et al. 2020] is still desirable. In the context of visual large language models, they might offer ways to tokenise the world based on visual content. Capturing individual objects is also desirable for physical reasoning, which is particularly important in robotics and constructing world models. Currently, either architectures aimed at the segmentation task or simple patch-based tokenisation are used instead due to simplicity in training such architectures. Compositional reconstruction models can be used in the real-world setting, yet several scaling challenges remain to make it as practical as alternatives.

In addition to exploring multi-object segmentation, CLEVRTEX has been adapted to disentangling objects' appearance variation factors [G. Singh et al. 2023] due to its diverse library of materials and shapes. Upon request from the community, we have compiled a new library of materials with permissive licenses to ensure that further work in this direction is supported.

7.1.2 Object Segmentation from Optical Flow

In chapter 3, we considered the problem of training a segmentation network by using optical flow as a supervision signal. As predicting optical flow from a single image is ambiguous, we approached this problem via motion anticipation, tasking a segmentation network with only predicting regions that are likely to move together (P3). To provide a learning signal, we devised a loss function that approximates the optical flow within a predicted region with a principled parametric model. We solved for the motion parameters in a closed form using a grouping of pixels (P1), training with the residual error as the objective. The idea is that an incorrect segmentation would introduce significant outliers in the flow reconstruction, increasing the residual error. As we only used optical flow during the loss computation stage, we could also apply the segmentation model to images. The approach established a new state-of-the-art across several benchmarks in motion and video object segmentation.

The work presented in chapter 3 influenced a series of follow-up works that pushed the boundaries of learning segmentation models with optical flow guidance. In [Lian et al. 2023; S. Singh et al. 2023], the authors proposed multi-stage self-training

pipelines that moderated the potential errors in optical flow by checking against appearance-based features. Our work has also been cited in the context of drone robotics [Fan et al. 2025]. The authors used more sophisticated parametric models fit using optimisation. In [Ranne et al. 2024], the authors explored GWM for catheter segmentation in ultrasound images and developed a related specialised flow-supervised segmentation approach. The work was also cited in a supervised setting [Xie et al. 2024] for a view of using RGB and motion information to inform the motion segmentation task.

7.1.3 Multi-object Segmentation with Motion Supervision

While principled and effective, our proposal in chapter 3 is somewhat limited to explaining scenes with a salient foreground object. One can usually reduce the GWM loss by using multiple components to explain the motion of one object. Multiple masks are required to predict a variable number of objects. The model could use any extra components to explain the same object whenever possible, resulting in over-segmentation. In chapter 4, we extended our guess-and-check approach to a multi-object setting to partially address this limitation. We considered a prior over possible transformations that approximately describe motion. This also allowed us to account for approximation errors of the assumed affine motion models and other inaccuracies by explicitly modelling noise. We showed empirically how this is beneficial in multi-object settings compared to GWM (chapter 3) and other methods that learn from optical flow.

In this work, we also introduced a new dataset that extends ClevrTex to videos by using a physics simulation. This enabled us to compare and study how motion can improve upon compositional approaches studied in chapter 2. We showed how our proposed loss can simply be added to existing appearance-based methods, enabling them to learn from motion and greatly improving their performance. Importantly, we showed how we can combine optical flow, our loss function and an off-the-shelf segmentation architecture to learn multi-object segmentation models for real-world data such as KITTI [Geiger et al. 2012] with state-of-the-art performance for unsupervised models.

Work in chapter 4 has influenced a series of follow-up works that explored motion-induced segmentation in multi-object settings. [Bao et al. 2023] considered an

object-centric slot model reconstructing flow. They addressed the limitation of slot-attention-based architectures with an external motion segmentation model used to guide the attention mechanism. In [Safadoust and Güney 2023], the authors jointly predicted depth and segmentation using optical flow for supervision. The estimate of depth enabled reasoning about motion in 3D, supporting multiple objects without a probabilistic framework.

7.1.4 Learning from Long-Range Motion

While showing strong results, our previous works only explored instantaneous motion as captured by optical flow. In chapter 5, we established the value of long-range motion information for unsupervised video segmentation, pioneering the use of point trajectories for this task in the deep learning era. Similar to our prior works in this area, we implement the learning principle as a loss function, avoiding any architectural constraints on the models. This enabled us to combine trajectory and flow information easily. With the use of long-range motion information, we were able to establish a new state-of-the-art on the challenging motion and video object segmentation benchmarks. While we used recent point tracking models to estimate the point trajectories, the generality of our loss function should allow extending it to any point-like constructs, e.g., joints or Gaussians [Kerbl et al. 2023].

7.1.5 Segmenting with Language

In chapter 6, we explored segmentation using existing structures in language to specify objects and delineate them. We followed the trend in computer vision and built upon large-scale foundational models to benefit from extensive data and compute resources dedicated to them. However, our proposal departed from the common approaches, as we did not fine-tune contrastively trained image-text encoders but rather repurposed a generative model without fine-tuning.

Language conditioning enables output control in generative models. We established how the mechanisms for adherence to language conditioning in diffusion architectures induce a way to represent and localise objects, already implementing the *naming* idea (P4). We showed that we can achieve state-of-the-art performance by adapting a diffusion model without fine-tuning. We, thus, showed how, by learning

a text-to-image generative model, one also learns an open-vocabulary segmentation model.

Our proposal has started or revitalised several trends in the field. [Barsellotti et al. 2024a; Barsellotti et al. 2024b] extended our work by addressing the problem of generating new prototypes for novel classes. Instead, the authors precomputed a database of various prototypes based on a corpus of prompts. Additionally, several works [S. Sun et al. 2024; Corradini et al. 2024] have explored how to adapt other vision-language foundation models for segmentation without training. Interestingly, our approach has been explored in the supervised settings as well. In [Zhao et al. 2025], the authors improved on our recipe to convert a diffusion model into a segmentation one by using LLMs to generate diverse prompts and incorporated SAM [Kirillov et al. 2023] to localise the objects.

7.1.6 Future Work

Curriculum and compositionality. Children learn by starting with simpler problems and then building up to more complex ones. We also break the problems down into smaller subtasks and consider simpler settings to build up intuition. However, compositional reconstruction models train to decompose cluttered scenes in one go from the start. Moreover, their formulation is often flexible to support changing the number of slots. It might be beneficial to consider a curriculum that starts with simpler scenes and gradually increases the complexity.

The power of *curriculum* has been shown in the self-labelling approaches to unsupervised segmentation [Van Gansbeke et al. 2021; Zadaianchuk et al. 2023b; Xinlong Wang et al. 2022; Xudong Wang et al. 2023a]. They start by training on object-centric datasets, which are less difficult to label with unsupervised saliency methods. During training, the segmentation model is heavily augmented with copy-paste augmentations [Ghiasi et al. 2021] to construct more complex scenes artificially. The label set is periodically updated with new objects, creating more targets for each scene. Following the same idea, one could exploit the generalisability of object-centric representations by learning them well on less cluttered scenes first before allowing for more slots and crowded scenes. Similarly, different augmented views of the scene could be employed to constrain the attention mechanism.

Learning from motion at scale. The models in chapters 3 to 5 have been trained on relatively small datasets using only small GPU resources. Much of modern success has been achieved at scale, so it is interesting to see how well the learning principles and loss functions would behave when increasing the model size and training on larger datasets. As the methods do not require annotations, abundant data is available.

Joint motion and appearance learning. Motion information is ultimately derived from the underlying video data. Preprocessing motion could be undesirable for several reasons. Firstly, the optical flow networks will not be perfect, and thus, the downstream segmentation model might be misled. Secondly, processing the video data to obtain motion information might prove cumbersome at scale due to associated computational and storage costs. One would prefer to learn from the video data directly without the need to preprocess it.

One could consider two networks trained jointly by taking inspiration from the two-stream hypothesis [Goodale and Milner 1992]. One would use the appearance-based principles, e.g. *compositional* reconstruction (P2), in the visual stream. For motion stream, one could use the self-supervised flow models [Stone et al. 2021]. The loss functions introduced in chapters 3 and 4 would bridge the two streams. This way, motion modelling could be informed by learning objects in appearance reconstruction, and appearance reconstruction could be informed by motion.

Language and motion. In chapter 6, we explored language and its understanding captured in generative diffusion models to segment objects. However, it is unclear whether language alone can provide sufficient information for a great segmentation, especially around boundaries. In the case of multiple instances of the same object class, language can be pretty ambiguous, segmenting multiple instances jointly. Motion, on the other hand, can help disambiguate between multiple instances. A potential approach to synergise these ideas (P3, P4) would be to distil our approach from chapter 6 to a segmentation model by pseudo-labelling a large video dataset. The same segmentation model could then also be supervised with motion information following schemes introduced in chapters 3 to 5.

Embracing object hierarchy with language. In this thesis, we have focused on learning objects at a defined granularity based on a choice of learning signal. Moving towards a more general hierarchical understanding of objects is a natural direction to enable better modelling of the world. Hierarchy is challenging, however, as not all levels are equally well-defined or interesting. One might produce superpixel-like parts of objects, which may or may not be meaningful. Language again can be helpful here, providing an existing structure to define meaningful hierarchy levels. One way to produce practical object hierarchies in the scene could be by leveraging the reasoning capabilities of recent large language models. For example, one could recursively prompt an LLM to provide relevant part descriptions for objects that could be then segmented, forming a deeper hierarchy level.

7.2 Conclusions

Learning to decompose the visual world into objects is critical for many tasks in computer vision. While much of the current art has focused on scaling supervised approaches, in this thesis, we have explored principles that can be used to learn objects without the need for costly, dense annotation. We drew inspiration from our understanding of human perception and language to examine and push the boundaries of unsupervised object segmentation and thus one day enable machines to understand the world the way we learn to.

References

Gilad Adiv (1985). "Determining three-dimensional motion and structure from optical flow generated by several moving objects". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel (2021). "Deep vit features as dense visual descriptors". In: *arXiv preprint arXiv:2112.05814*.

Nikita Araslanov and Stefan Roth (2020). "Single-stage semantic segmentation from image labels". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik (2010). "Contour detection and hierarchical image segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Pablo Arbelaez, Jordi Pont-Tuset, Jonathan T. Barron, Ferran Marques, and Jitendra Malik (June 2014). "Multiscale Combinatorial Grouping". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Shahaf Arica, Or Rubin, Sapir Gershov, and Shlomi Laufer (2024). "Cuvler: Enhanced unsupervised object discoveries through exhaustive self-supervised transformers". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Görkay Aydemir, Weidi Xie, and Fatma Guney (2024). "Self-supervised object-centric learning for videos". In: *Advances in Neural Information Processing Systems*.

Zhipeng Bao, Pavel Tokmakov, Allan Jabri, Yu-Xiong Wang, Adrien Gaidon, and Martial Hebert (June 2022). "Discovering Objects That Can Move". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Zhipeng Bao, Pavel Tokmakov, Yu-Xiong Wang, Adrien Gaidon, and Martial Hebert (2023). "Object discovery from motion-guided tokens". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko (2022). "Label-Efficient Semantic Segmentation with Diffusion Models". In: *International Conference on Learning Representations*.

Luca Barsellotti, Roberto Amoroso, Lorenzo Baraldi, and Rita Cucchiara (2024a). "Fossil: Free open-vocabulary semantic segmentation through synthetic references retrieval". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.

Luca Barsellotti, Roberto Amoroso, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara (2024b). "Training-Free Open-Vocabulary Segmentation with Offline Diffusion-Augmented Prototype Generation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Daniel Bear, Chaofei Fan, Damian Mrowca, Yunzhu Li, Seth Alter, Aran Nayebi, Jeremy Schwartz, Li Fei-Fei, Jiajun Wu, Josh Tenenbaum, and Daniel L. Yamins (2020). "Learning Physical Graph Representations from Visual Scenes". In: *Advances in Neural Information Processing Systems*.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling (2013). "The arcade learning environment: An evaluation platform for general agents". In: *Journal of Artificial Intelligence Research*.

Yaniv Benny and Lior Wolf (2020). "Onegan: Simultaneous unsupervised learning of conditional image generation, foreground segmentation, and fine-grained clustering". In: *Proceedings of the European Conference on Computer Vision*. Springer.

James R Bergen, Patrick Anandan, Keith J Hanna, and Rajesh Hingorani (1992). "Hierarchical model-based motion estimation". In: *Proceedings of the European Conference on Computer Vision*. Springer.

Beril Besbinar and Pascal Frossard (2021). "Self-Supervision by Prediction for Object Discovery in Videos". In: *2021 IEEE International Conference on Image Processing*.

Pia Bideau and Erik Learned-Miller (2016). "It's moving! a probabilistic model for causal motion segmentation in moving camera videos". In: *Proceedings of the European Conference on Computer Vision*. Springer.

Pia Bideau, Aruni RoyChowdhury, Rakesh R Menon, and Erik Learned-Miller (2018). "The best of both worlds: Combining cnns and geometric constraints for hierarchical motion segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Adam Bielski and Paolo Favaro (2019). "Emergence of object segmentation in perturbed generative models". In: *Advances in Neural Information Processing Systems*.

Adam Bielski and Paolo Favaro (2022). "MOVE: Unsupervised Movable Object Segmentation and Detection". In: *Advances in Neural Information Processing Systems*.

Ondrej Biza, Sjoerd van Steenkiste, Mehdi S. M. Sajjadi, Gamaleldin F. Elsayed, Aravindh Mahendran, and Thomas Kipf (2023). "Invariant Slot Attention: Object Discovery with Slot-Centric Reference Frames". In: *Proceedings of the International Conference on Machine Learning*.

Eran Borenstein and Shimon Ullman (2004). "Learning to segment". In: *Proceedings of the European Conference on Computer Vision*.

Richard Strong Bowen, Richard Tucker, Ramin Zabih, and Noah Snavely (2022). "Dimensions of Motion: Monocular Prediction through Flow Subspaces". In: *Proceedings of the International Conference on 3D Vision*.

Alican Bozkurt, Babak Esmaeili, Jennifer Dy, Dana Brooks, and Jan-Willem van de Meent (2019). *Tetrominoes dataset*. https://github.com/neu-pml/tetrominoes/.

Eric Brill (1992). "A simple rule-based part of speech tagger". In: *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Andrew Brock, Jeff Donahue, and Karen Simonyan (2019). "Large Scale GAN Training for High Fidelity Natural Image Synthesis". In.

Thomas Brox and Jitendra Malik (2010a). "Large displacement optical flow: descriptor matching in variational motion estimation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Thomas Brox and Jitendra Malik (2010b). "Object Segmentation by Long Term Analysis of Point Trajectories". In: *Proceedings of the European Conference on Computer Vision*.

Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez (2019). "Zero-shot semantic segmentation". In: *Advances in Neural Information Processing Systems*.

Chris Burgess and Hyunjik Kim (2018). *3D Shapes Dataset*. https://github.com/deepmind/3dshapes-dataset/.

Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner (2019). "Monet: Unsupervised scene decomposition and representation". In: *arXiv preprint arXiv:1901.11390*.

Serkan Cabi, Sergio Gómez Colmenarejo, Alexander Novikov, Ksenia Konyushkova, Scott Reed, Rae Jeong, Konrad Zolna, Yusuf Aytar, David Budden, Mel Vecerik,

et al. (2019). “Scaling data-driven robotics with reward sketching and batch reinforcement learning”. In: *Proceedings of Robotics: Science and Systems*.

Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool (2017). “One-shot video object segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari (2018). “COCO-Stuff: Thing and stuff classes in context”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze (2018). “Deep clustering for unsupervised learning of visual features”. In: *Proceedings of the European Conference on Computer Vision*.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin (2020). “Unsupervised Learning of Visual Features by Contrasting Cluster Assignments”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Curran Associates, Inc.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin (Oct. 2021). “Emerging Properties in Self-Supervised Vision Transformers”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Junbum Cha, Jonghwan Mun, and Byungseok Roh (2023). “Learning to generate text-grounded mask for open-world semantic segmentation from only image-text pairs”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Jason Chang and John W. Fisher III (2013). “Topology-Constrained Layered Tracking with Latent Flow”. In: *2013 IEEE International Conference on Computer Vision*.

Jeff Cheeger (1969). “A lower bound for the smallest eigenvalue of the Laplacian”. In: *Proceedings of the Princeton Conference in Honor of Professor S. Bochner*.

Honglin Chen, Rahul Venkatesh, Yoni Friedman, Jiajun Wu, Joshua B Tenenbaum, Daniel LK Yamins, and Daniel M Bear (2022). “Unsupervised Segmentation in Real-World Images via Spelke Object Inference”. In: *Proceedings of the European Conference on Computer Vision*.

Mickaël Chen, Thierry Artières, and Ludovic Denoyer (2019). “Unsupervised Object Segmentation by Redrawing”. In: *Advances in Neural Information Processing Systems*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton (2020). "A simple framework for contrastive learning of visual representations". In: *Proceedings of the International Conference on Machine Learning*. PMLR.

Vincent S. Chen, Paroma Varma, Ranjay Krishna, Michael Bernstein, Christopher Re, and Li Fei-Fei (2019). "Scene Graph Prediction With Limited Labels". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish V Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme Ruiz, Andreas Peter Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut (2023). "PaLI: A Jointly-Scaled Multilingual Language-Image Model". In: *Proceedings of the International Conference on Learning Representations*.

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar (2022). "Masked-attention mask transformer for universal image segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov (2021). "Per-Pixel Classification is Not All You Need for Semantic Segmentation". In: *Advances in Neural Information Processing Systems*.

Jiaxin Cheng, Soumyaroop Nandi, Prem Natarajan, and Wael Abd-Almageed (2021). "SIGN: Spatial-Information Incorporated Generative Network for Generalized Zero-Shot Semantic Segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu (2015). "Global Contrast Based Salient Region Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Anil M Cheriyadat and Richard J Radke (2009). "Non-negative matrix factorization of partial track data for motion segmentation". In: *2009 IEEE 12th International Conference on Computer Vision*.

Jang Hyun Cho, Utkarsh Mall, Kavita Bala, and Bharath Hariharan (2021). "Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Subhabrata Choudhury, Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht (2022). “Guess What Moves: Unsupervised Video and Image Segmentation by Anticipating Motion”. In: *Proceedings of the British Machine Vision Conference*.

Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi (2021). “Unsupervised Part Discovery from Contrastive Reconstruction”. In: *Advances in Neural Information Processing Systems*.

Kevin Clark and Priyank Jaini (2024). “Text-to-image diffusion models are zero shot classifiers”. In: *Advances in Neural Information Processing Systems*.

Dorin Comaniciu and Peter Meer (1999). “Mean shift analysis and applications”. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele (2016). “The Cityscapes Dataset for Semantic Urban Scene Understanding”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Barbara Toniella Corradini, Mustafa Shukor, Paul Couairon, Guillaume Couairon, Franco Scarselli, and Matthieu Cord (2024). “Freeseg-diff: Training-free open-vocabulary segmentation with diffusion models”. In: *arXiv preprint arXiv:2403.20105*.

Joao Costeira and Takeo Kanade (1995). “A multi-body factorization method for motion analysis”. In: *Proceedings of IEEE International Conference on Computer Vision*.

Joao Paulo Costeira and Takeo Kanade (1998). “A multibody factorization method for independently moving objects”. In: *International Journal of Computer Vision*.

Eric Crawford and Joelle Pineau (2019). “Spatially invariant unsupervised object detection with convolutional neural networks”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Eric Crawford and Joelle Pineau (2020). “Exploiting spatial invariance for scalable unsupervised object tracking”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. (2024). “Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models”. In: *arXiv preprint arXiv:2409.17146*.

Fei Deng, Zhuo Zhi, Donghun Lee, and Sungjin Ahn (2021). “Generative Scene Graph Networks”. In: *International Conference on Learning Representations*.

Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai (2022). "Decoupling zero-shot semantic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Shuangrui Ding, Weidi Xie, Yabo Chen, Rui Qian, Xiaopeng Zhang, Hongkai Xiong, and Qi Tian (2022). "Motion-inductive self-supervised object discovery in videos". In: *arXiv preprint arXiv:2210.00221*.

Andrea Dittadi, Samuele Papa, Michele De Vita, Bernhard Schölkopf, Ole Winther, and Francesco Locatello (2022). "Generalization and Robustness Implications in Object-Centric Learning". In: *Proceedings of the International Conference on Machine Learning*.

Carl Doersch, Abhinav Gupta, and Alexei A Efros (2015). "Unsupervised visual representation learning by context prediction". In: *Proceedings of the IEEE International Conference on Computer Vision*.

Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens Continente, Kucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang (2022). "TAP-Vid: A Benchmark for Tracking Any Point in a Video". In: *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*.

Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, João Carreira, et al. (2024). "Bootstap: Bootstrapped training for tracking-any-point". In: *Proceedings of the Asian Conference on Computer Vision*.

Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman (2023). "Tapir: Tracking any point with per-frame initialization and temporal refinement". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Carl Doersch and Andrew Zisserman (2017). "Multi-task self-supervised visual learning". In: *Proceedings of the IEEE International Conference on Computer Vision*.

A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox (2015). "FlowNet: Learning Optical Flow with Convolutional Networks". In: *Proceedings of the IEEE International Conference on Computer Vision*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. (2021). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*.

Ehsan Elhamifar and René Vidal (2013). "Sparse subspace clustering: Algorithm, theory, and applications". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Gamaleldin Elsayed, Aravindh Mahendran, Sjoerd Van Steenkiste, Klaus Greff, Michael C Mozer, and Thomas Kipf (2022). "Savi++: Towards end-to-end object-centric learning from real-world videos". In: *Advances in Neural Information Processing Systems*.

Patrick Emami, Pan He, Sanjay Ranka, and Anand Rangarajan (2021). "Efficient Iterative Amortized Inference for Learning Symmetric and Disentangled Multi-Object Representations". In: *Proceedings of the 38th International Conference on Machine Learning*. PMLR.

Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner (2021). "GENESIS-V2: Inferring Unordered Object Representations without Iterative Refinement". In: *Advances in Neural Information Processing Systems*.

Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner (2020). "Genesis: Generative scene inference and sampling with object-centric latent representations". In: *International Conference on Learning Representations*.

S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E. Hinton (2016). "Attend, Infer, Repeat: Fast Scene Understanding with Generative Models". In: *Proceedings of the International Conference on Neural Information Processing Systems*.

SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. (2018). "Neural scene representation and rendering". In: *Science*.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman (2012). *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman (2010). "The pascal visual object classes (voc) challenge". In: *International Journal of Computer Vision*.

Alon Faktor and Michal Irani (2014). "Video Segmentation by Non-Local Consensus voting". In: *Proceedings of the British Machine Vision Conference*.

Xuxiang Fan, Gongjian Wen, Zhinan Gao, Junlong Chen, and Haojun Jian (2025). "An Unsupervised Moving Object Detection Network for UAV Videos". In: *Drones*.

Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould (2017). "Self-supervised video representation learning with odd-one-out networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Matthieu Fradet, Philippe Robert, and Patrick Pérez (2009). "Clustering point trajectories with various life-spans". In: *Conference for Visual Media Production*.

Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, Megumi Sano, Kuno Kim, Elias Wang, Michael Lingelbach, Aidan Curtis, Kevin Tyler Feigelis, Daniel Bear, Dan Gutfreund, David Daniel Cox, Antonio Torralba, James J. DiCarlo, Joshua B. Tenenbaum, Josh Mcdermott, and Daniel LK Yamins (2021). "ThreeDWorld: A Platform for Interactive Multi-Modal Physical Simulation". In: *Advances in the Neural Information Processing Systems (Datasets and Benchmarks Track)*.

Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin Brualla, Pratul Srinivasan, Jonathan Barron, and Ben Poole (2025). "CAT3D: Create Anything in 3D with Multi-View Diffusion Models". In: *Advances in Neural Information Processing Systems*.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford (2018). "Datasheets for datasets". In: *arXiv preprint arXiv:1803.09010*.

Andreas Geiger, Philip Lenz, and Raquel Urtasun (2012). "Are we ready for autonomous driving? The KITTI vision benchmark suite". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph (2021). "Simple copy-paste is a strong data augmentation method for instance segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin (2022). "Scaling open-vocabulary image segmentation with image-level labels". In: *Proceedings of the European Conference on Computer Vision*.

J. J. Gibson (1950). *The Perception of the Visual World*. Houghton Mifflin, Boston.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis (2018). "Unsupervised Representation Learning by Predicting Image Rotations". In: *International Conference on Learning Representations*.

Rohit Girdhar and Deva Ramanan (2020). “CATER: A diagnostic dataset for Compositional Actions & TEmporal Reasoning”. In: *International Conference on Learning Representations*.

Melvyn A Goodale and A David Milner (1992). “Separate visual pathways for perception and action”. In: *Trends in Neurosciences*.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi (2022). “Kubric: a scalable dataset generator”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner (2019). “Multi-Object Representation Learning with Iterative Variational Inference”. In: *Proceedings of the International Conference on Machine Learning*.

Klaus Greff, Antti Rasmus, Mathias Berglund, Tele Hao, Harri Valpola, and Jürgen Schmidhuber (2016). “Tagger: Deep unsupervised perceptual grouping”. In: *Advances in Neural Information Processing Systems*.

Klaus Greff, Rupesh Kumar Srivastava, and Jürgen Schmidhuber (2015). “Binding via reconstruction clustering”. In: *arXiv preprint arXiv:1511.06418*.

Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber (2020). “On the binding problem in artificial neural networks”. In: *arXiv preprint arXiv:2012.05208*.

Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber (2017). “Neural expectation maximization”. In: *Proceedings of the International Conference on Neural Information Processing Systems*.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. (2020). “Bootstrap your own latent-a new approach to self-supervised learning”. In: *Advances in Neural Information Processing Systems*.

Oliver Groth, Fabian B Fuchs, Ingmar Posner, and Andrea Vedaldi (2018). "Shapestacks: Learning vision-based physical intuition for generalised object stacking". In: *Proceedings of the European Conference on Computer Vision*.

Zhangxuan Gu, Siyuan Zhou, Li Niu, Zihan Zhao, and Liqing Zhang (2020). "Context-aware feature generation for zero-shot semantic segmentation". In: *Proceedings of the 28th ACM International Conference on Multimedia*.

Laura Gustafson, Chloe Rolland, Nikhila Ravi, Quentin Duval, Aaron Adcock, Cheng-Yang Fu, Melissa Hall, and Candace Ross (2023). "Facet: Fairness in computer vision evaluation benchmark". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Benjamin David Haeffele, Chong You, and Rene Vidal (2020). "A Critique of Self-Expressive Deep Subspace Clustering". In: *International Conference on Learning Representations*.

Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T. Freeman (2022). "Unsupervised Semantic Segmentation by Distilling Feature Correspondences". In: *International Conference on Learning Representations*.

Tengda Han, Weidi Xie, and Andrew Zisserman (2019). "Video representation learning by dense predictive coding". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*.

Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki (2022). "Particle video revisited: Tracking through occlusions using point trajectories". In: *Proceedings of the European Conference on Computer Vision*. Springer.

R. I. Hartley and A. Zisserman (2004). *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick (2022). "Masked autoencoders are scalable vision learners". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick (2020). "Momentum contrast for unsupervised visual representation learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick (2017). "Mask r-cnn". In: *Proceedings of the IEEE International Conference on Computer Vision*.

Xingzhe He, Bastian Wandt, and Helge Rhodin (2022). "Ganseg: Learning to segment by unsupervised hierarchical image generation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Zhen He, Jian Li, Daxue Liu, Hangen He, and David Barber (2019). “Tracking by Animation: Unsupervised Learning of Multi-Object Attentive Trackers”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Olivier J. Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron van den Oord, Oriol Vinyals, and João Carreira (2021). “Efficient Visual Pretraining With Contrastive Detection”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Olivier J. Hénaff, Skanda Koppula, Evan Shelhamer, Daniel Zoran, Andrew Jaegle, Andrew Zisserman, João Carreira, and Relja Arandjelović (2022). “Object discovery and representation networks”. In: *Proceedings of the European Conference on Computer Vision*. Springer.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or (2023). “Prompt-to-prompt image editing with cross attention control”. In: *Proceedings of the International Conference on Learning Representations*.

Jonathan Ho, Ajay Jain, and Pieter Abbeel (2020). “Denoising diffusion probabilistic models”. In: *Advances in Neural Information Processing Systems*.

Jonathan Ho and Tim Salimans (2021). “Classifier-Free Diffusion Guidance”. In: *NeurIPS Workshop on Deep Generative Models and Downstream Applications*.

Berthold KP Horn and Brian G Schunck (1981). “Determining optical flow”. In: *Artificial Intelligence*.

Ronghang Hu, Shoubhik Debnath, Saining Xie, and Xinlei Chen (2022). “Exploring Long-Sequence Masked Autoencoders”. In: *arXiv preprint arXiv:2210.07224*.

Hsin-Ping Huang, Charles Herrmann, Junhwa Hur, Erika Lu, Kyle Sargent, Austin Stone, Ming-Hsuan Yang, and Deqing Sun (2023). “Self-supervised autoflow”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li (2022). “Flowformer: A transformer architecture for optical flow”. In: *Proceedings of the European Conference on Computer Vision*. Springer.

Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox (2017). “Flownet 2.0: Evolution of optical flow estimation with deep networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Allan Jabri, Andrew Owens, and Alexei Efros (2020). "Space-time correspondence as a contrastive random walk". In: *Advances in Neural Information Processing Systems*.

Suyog Jain, Bo Xiong, and Kristen Grauman (2017). "FusionSeg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos". In: *arXiv preprint arXiv:1701.05384*.

Eric Jang, Shixiang Gu, and Ben Poole (2017). "Categorical Reparameterization with Gumbel-Softmax". In: *Proceedings of the International Conference on Learning Representations*.

Allan D. Jepson and Michael J. Black (1993). "Mixture models for optical flow computation". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Xu Ji, Joao F Henriques, and Andrea Vedaldi (2019a). "Invariant information clustering for unsupervised image classification and segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Xu Ji, Joao F. Henriques, and Andrea Vedaldi (2019b). "Invariant Information Clustering for Unsupervised Image Classification and Segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig (2021). "Scaling up visual and vision-language representation learning with noisy text supervision". In: *Proceedings of the International Conference on Machine Learning*.

Jindong Jiang and Sungjin Ahn (2020). "Generative Neurosymbolic Machines". In: *Advances in Neural Information Processing Systems*.

Jindong Jiang, Fei Deng, Gautam Singh, and Sungjin Ahn (2023). "Object-centric slot diffusion". In: *Advances in Neural Information Processing Systems*.

Jindong Jiang, Sepehr Janghorbani, Gerard De Melo, and Sungjin Ahn (2020). "SCALOR: Generative World Models with Scalable Object Representations". In: *International Conference on Learning Representations*.

Longlong Jing, Yucheng Chen, and Yingli Tian (2019). "Coarse-to-fine semantic segmentation from image-level labels". In: *IEEE Transactions on Image Processing*.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick (2017). "CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Justin Johnson, Andrej Karpathy, and Li Fei-Fei (2016). "Densecap: Fully convolutional localization networks for dense captioning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

N. Jojic and B.J. Frey (2001). "Learning flexible sprites in video layers". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Rishabh Kabra, Daniel Zoran, Goker Erdogan, Loic Matthey, Antonia Creswell, Matthew Botvinick, Alexander Lerchner, and Christopher P Burgess (2021). "SIMONe: View-Invariant, Temporally-Abstracted Object Representations via Unsupervised Video Decomposition". In: *arXiv preprint arXiv:2106.03849*.

Asako Kanezaki (2018). "Unsupervised image segmentation by backpropagation". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht (2024). "Cotracker: It is better to track together". In: *Proceedings of the European Conference on Computer Vision*.

Laurynas Karazija, Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi (2022). "Unsupervised Multi-object Segmentation by Predicting Probable Motion Patterns". In: *Advances in Neural Information Processing Systems*.

Laurynas Karazija, Iro Laina, and Christian Rupprecht (2021). "ClevrTex: A Texture-Rich Benchmark for Unsupervised Multi-Object Segmentation". In: *Advances in the Neural Information Processing Systems (Datasets and Benchmarks Track)*.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis (2023). "3d gaussian splatting for real-time radiance field rendering." In: *ACM Trans. Graph.*

Margret Keuper (2017). "Higher-order minimum cost lifted multicuts for motion segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision*.

Margret Keuper, Bjoern Andres, and Thomas Brox (2015). "Motion Trajectory Segmentation via Minimum Cost Multicuts". In: *Proceedings of the IEEE International Conference on Computer Vision*.

Margret Keuper, Siyu Tang, Bjoern Andres, Thomas Brox, and Bernt Schiele (2020). "Motion Segmentation amp; Multiple Object Tracking by Correlation Co-Clustering". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Margret Keuper, Siyu Tang, Yu Zhongjie, Bjoern Andres, Thomas Brox, and Bernt Schiele (2016). "A multi-cut formulation for joint segmentation and tracking of multiple objects". In: *arXiv preprint arXiv:1607.06317*.

Chanyoung Kim, Woojung Han, Dayun Ju, and Seong Jae Hwang (2024). "EAGLE: Eigen Aggregation Learning for Object-Centric Unsupervised Semantic Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Jinwoo Kim, Janghyuk Choi, Ho-Jin Choi, and Seon Joo Kim (2023). "Shepherding slots to objects: Towards stable and robust object-centric learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Diederik P Kingma and Max Welling (2014). "Auto-encoding variational bayes". In: *International Conference on Learning Representations*.

Thomas Kipf, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff (2022). "Conditional Object-Centric Learning from Video". In: *Proceedings of the International Conference on Learning Representations*.

Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár (2019). "Panoptic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. (2023). "Segment anything". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Alexander Kolesnikov and Christoph H Lampert (2016). "Seed, expand and constrain: Three principles for weakly-supervised image segmentation". In: *Proceedings of the European Conference on Computer Vision*. Springer.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi (2017). "Ai2-thor: An interactive 3d environment for visual ai". In: *arXiv preprint arXiv:1712.05474*.

Adam R Kosiorek, Hyunjik Kim, Ingmar Posner, and Yee Whye Teh (2018). "Sequential Attend, Infer, Repeat: Generative Modelling of Moving Objects". In: *Advances in Neural Information Processing Systems*.

Adam R Kosiorek, Heiko Strathmann, Daniel Zoran, Pol Moreno, Rosalia Schneider, Soňa Mokrá, and Danilo J Rezende (2021). "NeRF-VAE: A Geometry Aware 3D

Scene Generative Model". In: *Proceedings of the International Conference on Machine Learning*.

Jannik Kossen, Karl Stelzner, Marcel Hussing, Claas Voelcker, and Kristian Kersting (2020). "Structured Object-Aware Physics Prediction for Video Modeling and Planning". In: *Proceedings of the International Conference on Learning Representations*.

Hala Lamdouar, Weidi Xie, and Andrew Zisserman (2021). "Segmenting Invisible Moving Objects". In: *Proceedings of the British Machine Vision Conference*.

Dong Lao, Zhengyang Hu, Francesco Locatello, Yanchao Yang, and Stefano Soatto (2023). "Divided Attention: Unsupervised Multi-Object Discovery with Contextually Separated Slots". In: *arXiv preprint arXiv:2304.01430*.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE*.

Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang (2017). "Unsupervised representation learning by sorting sequences". In: *Proceedings of the IEEE International Conference on Computer Vision*.

Alexander C Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak (2023). "Your diffusion model is secretly a zero-shot classifier". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl (2021). "Language-driven Semantic Segmentation". In: *International Conference on Learning Representations*.

Dylan Li and Gyungin Shin (2024). "ProMerge: Prompt and Merge for Unsupervised Instance Segmentation". In: *Proceedings of the European Conference on Computer Vision*.

Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg (2013). "Video Segmentation by Tracking Many Figure-Ground Segments". In: *Proceedings of the IEEE International Conference on Computer Vision*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi (2023). "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models". In: *Proceedings of the International Conference on Machine Learning*.

Nanbo Li, Cian Eastwood, and Robert Fisher (2020a). "Learning Object-Centric Representations of Multi-Object Scenes from Multiple Views". In: *Advances in Neural Information Processing Systems*.

Nanbo Li, Cian Eastwood, and Robert Fisher (2020b). "Learning object-centric representations of multi-object scenes from multiple views". In: *Advances in Neural Information Processing Systems*.

Peike Li, Yunchao Wei, and Yi Yang (2020). "Consistent structural relation learning for zero-shot segmentation". In: *Advances in Neural Information Processing Systems*.

Siyang Li, Bryan Seybold, Alexey Vorobyov, Alireza Fathi, Qin Huang, and C-C Jay Kuo (2018). "Instance embedding transfer to unsupervised video object segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Ziyi Li, Qinye Zhou, Xiaoyun Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie (2023). "Open-vocabulary object segmentation with diffusion models". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Long Lian, Zhirong Wu, and Stella X Yu (2023). "Bootstrapping Objectness from Videos by Relaxed Common Fate and Visual Grouping". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu (2023). "Open-vocabulary semantic segmentation with mask-adapted clip". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Ping-Sung Liao, Tse-Sheng Chen, Pau-Choo Chung, et al. (2001). "A fast algorithm for multilevel thresholding". In: *J. Inf. Sci. Eng.*

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick (2014). "Microsoft coco: Common objects in context". In: *Proceedings of the European Conference on Computer Vision*.

Zhixuan Lin, Yi-Fu Wu, Skand Peri, Bofeng Fu, Jindong Jiang, and Sungjin Ahn (2020a). "Improving generative imagination in object-centric world models". In: *Proceedings of the International Conference on Machine Learning*.

Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn (2020b). "SPACE: Unsupervised Object-Oriented Scene Representation via Spatial Attention and Decomposition". In: *International Conference on Learning Representations*.

Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma (2012). "Robust recovery of subspace structures by low-rank representation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen (2020). “Deep learning for generic object detection: A survey”. In: *International Journal of Computer Vision*.

Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang (June 2020). “Learning by Analogy: Reliable Supervision From Transformations for Unsupervised Optical Flow Estimation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Quande Liu, Youpeng Wen, Jianhua Han, Chunjing Xu, Hang Xu, and Xiaodan Liang (2022). “Open-world semantic segmentation via contrasting and clustering vision-language embedding”. In: *Proceedings of the European Conference on Computer Vision*.

Runtao Liu, Zhirong Wu, Stella Yu, and Stephen Lin (2021). “The Emergence of Objectness: Learning Zero-Shot Segmentation from Videos”. In: *Advances in Neural Information Processing Systems*.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo (2021). “Swin transformer: Hierarchical vision transformer using shifted windows”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf (2020). “Object-Centric Learning with Slot Attention”. In: *Advances in Neural Information Processing Systems*.

Ilya Loshchilov and Frank Hutter (2018). “Decoupled Weight Decay Regularization”. In: *Proceedings of the International Conference on Learning Representations*.

Can-Yi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan (2012). “Robust and efficient subspace segmentation via least squares regression”. In: *Proceedings of the European Conference on Computer Vision*.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu (2022). “Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models”. In: *arXiv preprint arXiv:2211.01095*.

Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli (2019). “See more, know more: Unsupervised video object segmentation with co-attention siamese networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Bruce D Lucas and Takeo Kanade (1981). "An iterative image registration technique with an application to stereo vision". In: *Proceedings of the International Joint Conference on Artificial Intelligence*.

Dijun Luo, Feiping Nie, Chris Ding, and Heng Huang (2011). "Multi-subspace representation and discovery". In: *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*.

Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li (2023). "SegCLIP: Patch Aggregation with Learnable Centers for Open-Vocabulary Semantic Segmentation". In: *Proceedings of the International Conference on Machine Learning*. PMLR.

Chaofan Ma, Yuhuan Yang, Chen Ju, Fei Zhang, Jinxiang Liu, Yu Wang, Ya Zhang, and Yanfeng Wang (2023). "DiffusionSeg: Adapting Diffusion Towards Unsupervised Object Discovery". In: *arXiv preprint arXiv:2303.09813*.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh (2017). "The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables". In: *Proceedings of the International Conference on Learning Representations*.

Aravindh Mahendran, James Thewlis, and Andrea Vedaldi (Sept. 2018). "Self-Supervised Segmentation by Grouping Optical-Flow". In: *Proceedings of the European Conference on Computer Vision Workshops*.

Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra (2019). "Habitat: A Platform for Embodied AI Research". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner (2017). *dSprites: Disentanglement testing Sprites dataset*. https://github.com/deepmind/dsprites-dataset/.

Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox (2016). "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi (June 2022a). "Deep Spectral Methods: A Surprisingly Strong Baseline for Unsupervised Semantic Segmentation and Localization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi (2022b). "Finding an Unsupervised Image Segmenter in each of your Deep Generative Models". In: *International Conference on Learning Representations*.

Etienne Meunier, Anaıš Badoual, and Patrick Bouthemy (2022). "Em-driven unsupervised learning for efficient motion segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Etienne Meunier and Patrick Bouthemy (2023a). "Segmenting the motion components of a video: A long-term unsupervised model". In: *arXiv preprint arXiv:2310.01040*.

Etienne Meunier and Patrick Bouthemy (2023b). "Unsupervised Space-Time Network for Temporally-Consistent Segmentation of Multiple Motions". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013). "Distributed representations of words and phrases and their compositionality". In: *Advances in Neural Information Processing Systems*.

Cheol-Hui Min, Jinseok Bae, Junho Lee, and Young Min Kim (2021). "GATSBI: Generative Agent-centric Spatio-temporal Object Interaction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Ishan Misra, C Lawrence Zitnick, and Martial Hebert (2016). "Shuffle and learn: unsupervised learning using temporal order verification". In: *Proceedings of the European Conference on Computer Vision*. Springer.

Tom Monnier, Elliot Vincent, Jean Ponce, and Mathieu Aubry (2021). "Unsupervised Layered Image Decomposition Into Object Prototypes". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille (2014). "The role of context for object detection and semantic segmentation in the wild". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Jishnu Mukhoti, Tsung-Yu Lin, Omid Poursaeed, Rui Wang, Ashish Shah, Philip HS Torr, and Ser-Nam Lim (2023). "Open vocabulary semantic segmentation with patch aligned contrastive learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Nazir Nayal, Misra Yavuz, Joao F Henriques, and Fatma Güney (2023). "Rba: Segmenting unknown regions rejected by all". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Duc Tam Nguyen, Maximilian Dax, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Zhongyu Lou, and Thomas Brox

(2019). “Deepusps: Deep robust unsupervised saliency prediction with self-supervision”. In: *Advances in Neural Information Processing Systems*.

Peter Ochs and Thomas Brox (2011). “Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions”. In: *Proceedings of the International Conference on Computer Vision*.

Peter Ochs and Thomas Brox (2012). “Higher order motion models and spectral clustering”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Peter Ochs, Jitendra Malik, and Thomas Brox (2014). “Segmentation of Moving Objects by Long Term Video Analysis”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

OpenAI (2023). *Introducing ChatGPT*. https://openai.com/blog/chatgpt.

Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic (2015). “Is object localization for free?-weakly-supervised learning with convolutional neural networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Lucas Prado Osco, Qiusheng Wu, Eduardo Lopes De Lemos, Wesley Nunes Gonçalves, Ana Paula Marques Ramos, Jonathan Li, and José Marcato Junior (2023). “The segment anything model (sam) for remote sensing applications: From zero to one shot”. In: *International Journal of Applied Earth Observation and Geoinformation*.

Yassine Ouali, Celine Hudelot, and Myriam Tami (2020). “Autoregressive Unsupervised Image Segmentation”. In: *Proceedings of the European Conference on Computer Vision*.

Anestis Papazoglou and Vittorio Ferrari (Dec. 2013). “Fast Object Segmentation in Unconstrained Video”. In: *Proceedings of the IEEE International Conference on Computer Vision*.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros (2016). “Context encoders: Feature learning by inpainting”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung (2016). “A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Georgy Ponimatkin, Nermin Samet, Yang Xiao, Yuming Du, Renaud Marlet, and Vincent Lepetit (2023). “A Simple and Powerful Global Optimization for Unsupervised Video Object Segmentation”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. (2021). "Learning transferable visual models from natural language supervision". In: *Proceedings of the International Conference on Machine Learning*.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen (2022). "Hierarchical text-conditional image generation with clip latents". In: *arXiv preprint arXiv:2204.06125*.

Kanchana Ranasinghe, Brandon McKinzie, Sachin Ravi, Yinfei Yang, Alexander Toshev, and Jonathon Shlens (2023). "Perceptual grouping in contrastive vision-language models". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Alex Ranne, Liming Kuang, Yordanka Velikova, Nassir Navab, and Ferdinando Rodriguez Y Baena (2024). "CathFlow: Self-Supervised Segmentation of Catheters in Interventional Ultrasound Using Optical Flow and Transformers". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Shankar R Rao, Roberto Tron, René Vidal, and Yi Ma (2008). "Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. (2024). "Sam 2: Segment anything in images and videos". In: *arXiv preprint arXiv:2408.00714*.

David P Reichert, Peggy Series, and Amos J Storkey (2011). "A hierarchical generative model of recurrent object-based attention in the visual cortex". In: *Proceedings of the International Conference on Artificial Neural Networks*.

Pengzhen Ren, Changlin Li, Hang Xu, Yi Zhu, Guangrun Wang, Jianzhuang Liu, Xiaojun Chang, and Xiaodan Liang (2023). "ViewCo: Discovering Text-Supervised Segmentation Masks via Multi-View Semantic Consistency". In: *The Eleventh International Conference on Learning Representations*.

Danilo Jimenez Rezende and Fabio Viola (2018). "Taming VAEs". In: *arXiv preprint arXiv:1810.00597*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer (2022). "High-resolution image synthesis with latent diffusion models". In:

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. (2015). "Imagenet large scale visual recognition challenge". In: *International Journal of Computer Vision.*

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton (2017). "Dynamic routing between capsules". In: *Proceedings of the International Conference on Neural Information Processing Systems.*

Sadra Safadoust and Fatma Güney (2023). "Multi-object discovery by low-dimensional object motion". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.*

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. (2022). "Photorealistic text-to-image diffusion models with deep language understanding". In: *Advances in Neural Information Processing Systems.*

Mohammadreza Salehi, Efstratios Gavves, Cees GM Snoek, and Yuki M Asano (2023). "Time does tell: Self-supervised time-tuning of dense image representations". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.*

Peter Sand and Seth Teller (2008). "Particle video: Long-range motion estimation using point trajectories". In: *International Journal of Computer Vision.*

Manolis Savva, Angel X. Chang, Alexey Dosovitskiy, Thomas Funkhouser, and Vladlen Koltun (2017). "MINOS: Multimodal Indoor Simulator for Navigation in Complex Environments". In: *arXiv:1712.03931.*

Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting (2023). "Safe Latent Diffusion: Mitigating Inappropriate Degeneration in Diffusion Models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.*

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. (2022). "Laion-5b: An open large-scale dataset for training

next generation image-text models". In: *Advances in Neural Information Processing Systems*.

Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello (2023). "Bridging the Gap to Real-World Object-Centric Learning". In: *Proceedings of the International Conference on Learning Representations*.

Hyun Seok Seong, WonJun Moon, SuBeen Lee, and Jae-Pil Heo (2023). "Leveraging hidden positives for unsupervised semantic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut (2018). "Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning". In: *Proceedings of the Association for Computational Linguistics*.

Jianbo Shi and Jitendra Malik (2000). "Normalized cuts and image segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia (2016). "Hierarchical Image Saliency Detection on Extended CSSD". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Gyungin Shin, Samuel Albanie, and Weidi Xie (June 2022a). "Unsupervised Salient Object Detection With Spectral Cluster Voting". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*.

Gyungin Shin, Weidi Xie, and Samuel Albanie (2021). "All you need are a few pixels: semantic segmentation with pixelpick". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Gyungin Shin, Weidi Xie, and Samuel Albanie (2022b). "ReCo: Retrieve and Co-segment for Zero-shot Transfer". In: *Advances in Neural Information Processing Systems*.

Oriane Siméoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce (Nov. 2021). "Localizing Objects with Self-Supervised Transformers and no Labels". In: *Proceedings of the British Machine Vision Conference*.

Oriane Siméoni, Chloé Sekkat, Gilles Puy, Antonıı Vobeckỳ, Éloi Zablocki, and Patrick Pérez (2023). "Unsupervised object localization: Observing the background to discover objects". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman (2013). "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint arXiv:1312.6034*.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela (2022). "Flava: A foundational language and vision alignment model". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Gautam Singh, Sungjin Ahn, and Yeongbin Kim (2023). "Neural Systematic Binder". In: *Proceedings of the International Conference on Learning Representations*. The International Conference on Learning Representations (ICLR).

Gautam Singh, Fei Deng, and Sungjin Ahn (2022a). "Illiterate DALL-E Learns to Compose". In: *Proceedings of the International Conference on Learning Representations*.

Gautam Singh, Skand Peri, Junghyun Kim, Hyunseok Kim, and Sungjin Ahn (2021). "Structured world belief for reinforcement learning in POMDP". In: *Proceedings of the International Conference on Machine Learning*. PMLR.

Gautam Singh, Yi-Fu Wu, and Sungjin Ahn (2022b). "Simple unsupervised object-centric learning for complex and naturalistic videos". In: *Advances in Neural Information Processing Systems*.

Silky Singh, Shripad Deshmukh, Mausoom Sarkar, and Balaji Krishnamurthy (2023). "LOCATE: Self-supervised Object Discovery via Flow-guided Graph-cut and Bootstrapped Self-training". In: *Proceedings of the British Machine Vision Conference*.

Dmitriy Smirnov, Michael Gharbi, Matthew Fisher, Vitor Guizilini, Alexei A Efros, and Justin Solomon (2021). "MarioNette: Self-Supervised Sprite Learning". In: *Advances in Neural Information Processing Systems*.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli (2015). "Deep unsupervised learning using nonequilibrium thermodynamics". In: *Proceedings of the International Conference on Machine Learning*.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole (2021). "Score-Based Generative Modeling through Stochastic Differential Equations". In: *Proceedings of the International Conference on Learning Representations*.

Elizabeth S Spelke (1990). "Principles of object perception". In: *Cognitive Science*.

Elizabeth S Spelke and Katherine D Kinzler (2007). "Core knowledge". In: *Developmental Science*.

Karl Stelzner, Kristian Kersting, and Adam R Kosiorek (2021). “Decomposing 3D Scenes into Objects via Unsupervised Volume Segmentation”. In: *arXiv preprint arXiv:2104.01148*.

Karl Stelzner, Robert Peharz, and Kristian Kersting (2019). “Faster Attend-Infer-Repeat with Tractable Probabilistic Models”. In: *Proceedings of the 36th International Conference on Machine Learning*. PMLR.

Austin Stone, Daniel Maurer, Alper Ayvaci, Anelia Angelova, and Rico Jonschkowski (2021). “SMURF: Self-Teaching Multi-Frame Unsupervised RAFT with Full-Image Warping”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz (June 2018). “PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Shuyang Sun, Runjia Li, Philip Torr, Xiuye Gu, and Siyang Li (2024). “Clip as rnn: Segment countless visual concepts without training endeavor”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Narayanan Sundaram, Thomas Brox, and Kurt Keutzer (2010). “Dense Point Trajectories by GPU-Accelerated Large Displacement Optical Flow”. In: *Proceedings of the European Conference on Computer Vision*.

Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Ture (2023). “What the DAAM: Interpreting Stable Diffusion Using Cross Attention”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Zachary Teed and Jia Deng (2020). “RAFT: Recurrent All-Pairs Field Transforms for Optical Flow”. In: *Proceedings of the European Conference on Computer Vision*.

Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid (2017). “Learning video object segmentation with visual memory”. In: *Proceedings of the IEEE International Conference on Computer Vision*.

Pavel Tokmakov, Cordelia Schmid, and Karteek Alahari (Mar. 2019). “Learning to Segment Moving Objects”. In: *International Journal of Computer Vision*.

Carlo Tomasi and Takeo Kanade (1991). “Detection and tracking of point”. In: *Int J Comput Vis*.

Philip H. S. Torr (1998). “Geometric motion segmentation and model selection”. In: *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*.

James Townsend (2016). “Differentiating the Singular Value Decomposition”. In: URL: https://j-towns.github.io/papers/svd-derivative.pdf.

Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J. Black (2016). “Video Segmentation via Object Flow”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Vishaal Udandarao, Ankush Gupta, and Samuel Albanie (2023). “Sus-x: Training-free name-only transfer of vision-language models”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky (2020). “Deep Image Prior”. In: *IJCV*.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool (2020). “Scan: Learning to classify images without labels”. In: *Proceedings of the European Conference on Computer Vision*. Springer.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool (2021). “Unsupervised Semantic Segmentation by Contrasting Object Mask Proposals”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Sjoerd Van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber (2018). “Relational neural expectation maximization: Unsupervised discovery of objects and their interactions”. In: *arXiv preprint arXiv:1802.10353*.

Rishi Veerapaneni, John D Co-Reyes, Michael Chang, Michael Janner, Chelsea Finn, Jiajun Wu, Joshua Tenenbaum, and Sergey Levine (2020). “Entity abstraction in visual model-based reinforcement learning”. In: *Conference on Robot Learning*. PMLR.

René Vidal and Paolo Favaro (2014). “Low rank subspace clustering (LRSC)”. In: *Pattern Recognition Letters*.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol (2008). “Extracting and composing robust features with denoising autoencoders”. In: *Proceedings of the International Conference on Machine Learning*.

Andrey Voynov, Stanislav Morozov, and Artem Babenko (2020). “Big gans are watching you: Towards unsupervised object segmentation with off-the-shelf generative models”. In: *arXiv preprint arXiv:2006.04988*.

Andrey Voynov, Stanislav Morozov, and Artem Babenko (2021). “Object segmentation without labels with large-scale generative models”. In: *Proceedings of the International Conference on Machine Learning*. PMLR.

Johan Wagemans, Jacob Feldman, Sergei Gepshtein, Ruth Kimchi, James R Pomerantz, Peter A Van der Helm, and Cees Van Leeuwen (2012). "A century of Gestalt psychology in visual perception: II. Conceptual and theoretical foundations." In: *Psychological Bulletin*.

Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan (2017). "Learning to Detect Salient Objects With Image-Level Supervision". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Wenguan Wang, Jianbing Shen, Ruigang Yang, and Fatih Porikli (2018). "Saliency-Aware Video Object Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xiaolong Wang, Allan Jabri, and Alexei A Efros (2019). "Learning correspondence from the cycle-consistency of time". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M Alvarez (2022). "Freesolo: Learning to segment objects without annotations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li (2021). "Dense contrastive learning for self-supervised visual pre-training". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra (2023a). "Cut and learn for unsupervised object detection and instance segmentation". In: *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*.

Xudong Wang, Ishan Misra, Ziyun Zeng, Rohit Girdhar, and Trevor Darrell (2023b). "VideoCutLER: Surprisingly Simple Unsupervised Video Instance Segmentation". In: *arXiv preprint arXiv:2308.14710*.

Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L. Crowley, and Dominique Vaufreydaz (2022). "Self-supervised Transformers for Unsupervised Object Discovery using Normalized Cut". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Ziyu Wang, Mike Zheng Shou, and Mengmi Zhang (2023). "Object-centric learning with cyclic walks between parts and whole". In: *Advances in Neural Information Processing Systems*.

Nicholas Watters, Loic Matthey, Matko Bosnjak, Christopher P Burgess, and Alexander Lerchner (2019a). "Cobra: Data-efficient model-based rl through

unsupervised object discovery and curiosity-driven exploration". In: *arXiv preprint arXiv:1905.09275*.

Nicholas Watters, Loic Matthey, Christopher P Burgess, and Alexander Lerchner (2019b). "Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes". In: *arXiv preprint arXiv:1901.07017*.

Nicholas Watters, Daniel Zoran, Theophane Weber, Peter Battaglia, Razvan Pascanu, and Andrea Tacchetti (2017). "Visual interaction networks: Learning a physics simulator from video". In: *Advances in Neural Information Processing Systems*.

Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman (2018). "Learning and using the arrow of time". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Yichen Wei, Fang Wen, Wangjiang Zhu, and Jian Sun (2012). "Geodesic Saliency Using Background Priors". In: *Proceedings of the European Conference on Computer Vision*.

Marissa A Weis, Kashyap Chitta, Yash Sharma, Wieland Brendel, Matthias Bethge, Andreas Geiger, and Alexander S Ecker (2021). "Benchmarking unsupervised object representations for video sequences". In: *Journal of Machine Learning Research*.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona (2010). *Caltech-UCSD Birds 200*. Tech. rep. California Institute of Technology.

Max Wertheimer (1912). "Experimentelle studien uber das sehen von bewegung". In: *Zeitschrift fur psychologie*.

Benjamin Wilson, Judy Hoffman, and Jamie Morgenstern (2019). "Predictive inequity in object detection". In: *arXiv preprint arXiv:1902.11097*.

Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen (2023). "Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Yizhe Wu, Oiwi Parker Jones, Martin Engelcke, and Ingmar Posner (2021). "APEX: Unsupervised, Object-Centric Scene Segmentation and Tracking for Robot Manipulation". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg (2023). "SlotFormer: Unsupervised Visual Dynamics Simulation with Object-Centric Models". In: *Proceedings of the International Conference on Learning Representations*.

Monika Wysoczańska, Michaël Ramamonjisoa, Tomasz Trzciński, and Oriane Siméoni (2024). "Clip-diy: Clip dense inference yields open-vocabulary semantic segmentation

for-free". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.

Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese (2018). "Gibson env: Real-world perception for embodied agents". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Xide Xia and Brian Kulis (2017). "W-net: A deep model for fully unsupervised image segmentation". In: *arXiv preprint arXiv:1711.08506*.

Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata (2019). "Semantic projection network for zero-and few-label semantic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Junyu Xie, Weidi Xie, and Andrew Zisserman (2022). "Segmenting moving objects via an object-centric layered representation". In: *Advances in Neural Information Processing Systems*.

Junyu Xie, Charig Yang, Weidi Xie, and Andrew Zisserman (2024). "Moving Object Segmentation: All You Need Is SAM (and Flow)". In: *Proceedings of the Asian Conference on Computer Vision*.

Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei (2017). "Scene graph generation by iterative message passing". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang (2022). "Groupvit: Semantic segmentation emerges from text supervision". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello (2023). "Open-vocabulary panoptic segmentation with text-to-image diffusion models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Jilan Xu, Junlin Hou, Yuejie Zhang, Rui Feng, Yi Wang, Yu Qiao, and Weidi Xie (2023). "Learning open-vocabulary semantic segmentation models from natural language supervision". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Kun Xu, Chongxuan Li, Jun Zhu, and Bo Zhang (2019). "Multi-Object Generation with Amortized Structural Regularization". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc.

Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai (2022). "A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model". In: *Proceedings of the European Conference on Computer Vision*.

Jingyu Yan and Marc Pollefeys (2006). "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate". In: *Proceedings of the European Conference on Computer Vision*. Springer.

Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie (2021). "Self-supervised video object segmentation by motion grouping". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang (2013). "Saliency detection via graph-based manifold ranking". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao (2023). "Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v". In: *arXiv preprint arXiv:2310.11441*.

Yanchao Yang, Yutong Chen, and Stefano Soatto (2020). "Learning to manipulate individual objects in an image". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yanchao Yang, Brian Lai, and Stefano Soatto (2021). "Dystab: Unsupervised object segmentation via dynamic-static bootstrapping". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto (June 2019). "Unsupervised Moving Object Detection via Contextual Information Separation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Vickie Ye, Zhengqi Li, Richard Tucker, Angjoo Kanazawa, and Noah Snavely (June 2022). "Deformable Sprites for Unsupervised Video Decomposition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Hong-Xing Yu, Leonidas J. Guibas, and Jiajun Wu (2021). "Unsupervised Discovery of Object Radiance Fields". In: *arXiv preprint arXiv:2107.07905*.

Sukmin Yun, Seong Hyeon Park, Paul Hongsuck Seo, and Jinwoo Shin (2023). "Ifseg: Image-free semantic segmentation via vision-language model". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Polina Zablotskaia, Edoardo A Dominici, Leonid Sigal, and Andreas M Lehrmann (2020). "Unsupervised video decomposition using spatio-temporal iterative inference". In: *arXiv preprint arXiv:2006.14727*.

Polina Zablotskaia, Edoardo A. Dominici, Leonid Sigal, and Andreas M. Lehrmann (2021). "PROVIDE: a probabilistic framework for unsupervised video decomposition". In: *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.

Andrii Zadaianchuk, Matthaeus Kleindessner, Yi Zhu, Francesco Locatello, and Thomas Brox (2023a). "Unsupervised Semantic Segmentation with Self-supervised Object-centric Representations". In: *Proceedings of the International Conference on Learning Representations*.

Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius (2023b). "Object-centric learning for real-world videos by predicting temporal feature similarities". In: *Advances in Neural Information Processing Systems*.

Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius (2023c). "Self-supervised Visual Reinforcement Learning with Object-centric Representations". In: *Proceedings of the International Conference on Learning Representations*.

Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius (2024). "Object-centric learning for real-world videos by predicting temporal feature similarities". In: *Advances in Neural Information Processing Systems*.

Andy Zeng, Kuan-Ting Yu, Shuran Song, Daniel Suo, Ed Walker, Alberto Rodriguez, and Jianxiong Xiao (2017). "Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge". In: *Proceedings of the IEEE International Conference on Robotics and Automation*.

Yu Zeng, Yunzhi Zhuge, Huchuan Lu, Lihe Zhang, Mingyang Qian, and Yizhou Yu (2019). "Multi-source weak supervision for saliency detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer (2023). "Sigmoid loss for language image pre-training". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Han Zhang, Fangyi Chen, Zhiqiang Shen, Qiqi Hao, Chenchen Zhu, and Marios Savvides (2020). "Solving missing-annotation object detection with background recalibration loss". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*.

Jing Zhang, Tong Zhang, Yuchao Dai, Mehrtash Harandi, and Richard Hartley (2018). "Deep unsupervised saliency detection: A multiple noisy labeling perspective". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Yu Qiao, Peng Gao, and Hongsheng Li (2023). "Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Richard Zhang, Phillip Isola, and Alexei A Efros (2016). "Colorful image colorization". In: *Proceedings of the European Conference on Computer Vision*.

Xiao Zhang and Michael Maire (2020). "Self-supervised visual representation learning from hierarchical grouping". In: *Advances in Neural Information Processing Systems*.

Xuanpu Zhao, Dianmo Sheng, Zhentao Tan, Zhiwei Zhao, Tao Gong, Qi Chu, Bin Liu, and Nenghai Yu (2025). "Training-free Open-Vocabulary Semantic Segmentation via Diverse Prototype Construction and Sub-region Matching". In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas (2023). "PointOdyssey: A Large-Scale Synthetic Dataset for Long-Term Point Tracking". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba (2016). "Learning deep features for discriminative localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba (2017). "Scene parsing through ade20k dataset". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Chong Zhou, Chen Change Loy, and Bo Dai (2022). "Extract free dense labels from clip". In: *Proceedings of the European Conference on Computer Vision*.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong (2021). "Image BERT Pre-training with Online Tokenizer". In: *International Conference on Learning Representations*.

Adrian Ziegler and Yuki M Asano (2022). "Self-supervised learning of object parts for semantic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Daniel Zoran, Rishabh Kabra, Alexander Lerchner, and Danilo J Rezende (2021). "Parts: Unsupervised segmentation with slots, attention and independence maximization". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee (2023). "Segment everything everywhere all at once". In: *Advances in Neural Information Processing Systems*.

Appendix A

Statement of Authorship

A statement of authorship is provided for each multi-authored paper included in this thesis. The statements describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication, there exists a complete statement that is filled out and signed by the candidate and supervisor.

Statement of Authorship for the paper "ClevrTex: A Texture-Rich Benchmark for Unsuper- vised Multi-Object Segmentation" in Chapter 2.

Paper title	ClevrTex: A Texture-Rich Benchmark for Unsupervised Multi-Object Segmentation
Authors	Laurynas Karazija, Iro Laina, Christian Rupprecht
Publication status	Published
Publication details	In Conference on Neural Information Processing Systems (NeurIPS) 2021 Datasets and Benchmarks Track.

Student Confirmation

Student name	Laurynas Karazija	
Contribution to the paper	First-author contribution: • Conception of the research idea • Design and production of the ClevrTex datasets • Implementation of the methods, running of the experiments • Writing and presentation of the paper	
Signature and Date	Laurynas Karazija	April 19th 2025

Supervisor Confirmation

By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Professor Andrea Vedaldi	
Supervisor comments	The description of the contributions is accurate.	
Signature and Date	Andrea Vedaldi	April 21st 2025

Statement of Authorship for the paper "Guess What Moves: Unsupervised Video and Image Segmentation by Anticipating Motion" in Chapter 3.

Paper title	Guess What Moves: Unsupervised Video and Image Segmentation by Anticipating Motion
Authors	Subhabrata Choudhury*, Laurynas Karazija*, Iro Laina, Andrea Vedaldi, Christian Rupprecht * indicates equal contribution
Publication status	Published
Publication details	In British Machine Vision Conference (BMVC), 2022

Student Confirmation

Student name	Laurynas Karazija	
Contribution to the paper	First-author contribution: • Development of the loss function for quadratic formulation • Design and running of the experiments, ablation studies • Conception of the merging strategy • Writing and presentation of the paper	
Signature and Date	Laurynas Karazija	April 19th 2025

Supervisor Confirmation

By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Professor Andrea Vedaldi	
Supervisor comments	The description of the contributions is accurate.	
Signature and Date	Andrea Vedaldi	April 21st 2025

Statement of Authorship for the paper "Unsupervised Multi-object Segmentation by Predicting Probable Motion Patterns" in Chapter 4.

Paper title	Unsupervised Multi-object Segmentation by Predicting Probable Motion Patterns
Authors	Laurynas Karazija*, Subhabrata Choudhury*, Iro Laina, Christian Rupprecht, Andrea Vedaldi * indicates equal contribution
Publication status	Published
Publication details	In Advances in Neural Information Processing Systems, 2022

Student Confirmation

Student name	Laurynas Karazija	
Contribution to the paper	First-author contribution: • Deriving the loss function • Design and running of the experiments and ablation studies • Making of the Moving CLEVRTEX dataset • Writing and presentation of the paper	
Signature and Date	Laurynas Karazija	April 19th 2025

Supervisor Confirmation

By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Professor Andrea Vedaldi	
Supervisor comments	The description of the contributions is accurate.	
Signature and Date	Andrea Vedaldi	April 21st 2025

Statement of Authorship for the paper "Learning segmentation from point trajectories" in Chapter 5.

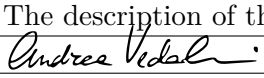
Paper title	Learning segmentation from point trajectories
Authors	Laurynas Karazija, Iro Laina, Christian Rupprecht, Andrea Vedaldi
Publication status	Published
Publication details	In Advances in Neural Information Processing Systems, 2024

Student Confirmation

Student name	Laurynas Karazija	
Contribution to the paper	First-author contribution: • Conception, derivation and implementation of the loss function • Design and running of the experiments and ablation studies • Writing and presentation of the paper	
Signature and Date		April 19th 2025

Supervisor Confirmation

By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Professor Andrea Vedaldi	
Supervisor comments	The description of the contributions is accurate.	
Signature and Date		April 21st 2025

Statement of Authorship for the paper "Diffusion Models for Open-Vocabulary Segmentation" in Chapter 6.

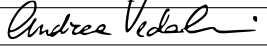
Paper title	Diffusion Models for Open-Vocabulary Segmentation
Authors	Laurynas Karazija, Iro Laina, Andrea Vedaldi, Christian Rupprecht
Publication status	Published
Publication details	In European Conference on Computer Vision (ECCV), 2024

Student Confirmation

Student name	Laurynas Karazija	
Contribution to the paper	First-author contribution: • Conception of the research idea • Development and execution of the experiments • Writing and presentation of the paper	
Signature and Date		April 19th 2025

Supervisor Confirmation

By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Professor Andrea Vedaldi	
Supervisor comments	The description of the contributions is accurate.	
Signature and Date		April 21st 2025

Appendix B

ClevrTex: A Texture-Rich Benchmark for Unsupervised Multi-Object Segmentation Supplementary Material

B.1 Dataset Documentation: Datasheets for Datasets

Here we answer the questions outlined in the datasheets for datasets paper by [Gebru et al. 2018].

B.1.1 Motivation

For what purpose was the dataset created? CLEVRTEX was created to serve as the next challenging benchmark for unsupervised multi-object segmentation methods. It trades simpler visuals for confounding aspects such as texture, irregular shapes, and a variety of materials.

Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organisation)? The dataset has been constructed by the research group "Visual Geometry Group" at the Engineering Science Department, University of Oxford.

Who funded the creation of the dataset? The dataset is created for research purposes at VGG. L. K. is funded by EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems EP/S024050/1. I. L. is supported by the EPSRC programme grant Seebibyte EP/M013774/1 and ERC starting grant IDIU-638009. C. R. is supported by Innovate UK (project 71653) on behalf of UK Research and Innovation (UKRI) and by the European Research Council (ERC) IDIU-638009.

B.1.2 Composition

What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? The dataset consists of images featuring simulated scenes and segmentation, depth, normal, albedo, and shadow masks available, and metadata detailing scene composition.

How many instances are there in total (of each type, if appropriate)? There are 50 000 instances in the main CLEVRTEX dataset. 20 000 in each variant, PLAINBG, VARBG, GRASSBG and CAMO. There is also a further 10 000 instances in the testing-only variant OOD.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? The dataset is a sample of the near-infinite set of possible arrangements under our sampling distribution. Please see section 2.3.1 for a description of the process to sample the scene.

What data does each instance consist of? Each instance consists of the RGB scene image, depth, normal, albedo, and shadow masks (all PNG), and further metadata (JSON) detailing object positions, shapes, scales, and materials used. We use only the RBG image for training during the benchmarking process and segmentation masks and metadata to evaluate.

Is there a label or target associated with each instance? For the task explored in this paper, unsupervised multi-object segmentation, the target labels are the segmentation masks, which are not used during training.

Is any information missing from individual instances? No.

Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? No, there are no relationships between different instances.

Are there recommended data splits (e.g., training, development/validation, testing)? Yes, we adopt 10%/10%/80% test/val/train splits for the datasets by instance index, with the exception of OOD variant, which is used for evaluation only. The rationale behind splits is that the data comes from the same generation process for each variant and can already be considering randomized. Simply using an image index to separate the splits makes both data-loading easy and removes the need to distribute canonical split indexes.

Are there any errors, sources of noise, or redundancies in the dataset? No.

Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? The dataset is self-contained.

Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals' non-public communications)? No.

Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? No.

Does the dataset relate to people? If not, you may skip the remaining questions in this section. No.

Does the dataset identify any subpopulations (e.g., by age, gender)? NA

Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset? NA

Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)? NA

B.1.3 Collection process

How was the data associated with each instance acquired? The data was generated.

What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)? The images were rendered using Blender 2.9.3 software on generic systems.

If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)? See the similar question in the Composition section.

Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)? The authors were involved in the process of generating this dataset.

Over what timeframe was the data collected? The datasets were rendered over a period of several weeks.

Were any ethical review processes conducted (e.g., by an institutional review board)? No.

Does the dataset relate to people? If not, you may skip the remainder of the questions in this section. No.

B.1.4 Preprocessing/cleaning/labeling

Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? No, the dataset was generated together with labels.

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? NA

Is the software used to preprocess/clean/label the instances available? NA

B.1.5 Uses

Has the dataset been used for any tasks already? In the paper we show and benchmark the intended use of this dataset for unsupervised multi-object segmentation setting.

Is there a repository that links to any or all papers or systems that use the dataset? We will be listing these on the website.

What (other) tasks could the dataset be used for? We include additional information maps when generating this dataset, which could be used for exploring value of using extra modalities for supervision or as targets. As mentioned before, we also generated necessary metadata for CLEVR-like QA task.

Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? No.

Are there tasks for which the dataset should not be used? This dataset is meant for research purposes only.

B.1.6 Distribution

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created? No.

How will the dataset will be distributed (e.g., tarball on website, API, GitHub)? The dataset and related evaluation code is available on the website https://www.robots.ox.ac.uk/~vgg/research/clevrtex/ allowing users to download and read-in the data.

When will the dataset be distributed? The dataset is available now.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? CC-BY.

Have any third parties imposed IP-based or other restrictions on the data associated with the instances? The original textures used in rendering objects are copyrighted by Poliigon Pty Ltd and cannot be redistributed to a third party. This only applies to texture images used in creating this dataset. The materials used for main dataset are freely available under non-commercial license and we include instructions to retrieve them alongside the generation code. Textures used in evaluation-only OOD variant are not available free of charge (we obtained them under a commercial license), but their catalogue is similarly included with the code. The dataset instances themselves do not have IP-based restrictions.

Do any export controls or other regulatory restrictions apply to the dataset or to individual instances? Not that we are are of. Regular UK laws apply.

B.1.7 Maintenance

Who is supporting/hosting/maintaining the dataset? The dataset is supported by the authors and by the VGG research group. The main contact person

is Laurynas Karazija.

How can the owner/curator/manager of the dataset be contacted (e.g., email address)? The authors of this dataset can be reached at their e-mail addresses: *{laurynas,chrisr,iro}@robots.ox.ac.uk*.

Is there an erratum? If errors are found and erratum will be added to the website.

Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? Any potential future updates or extension will be communicated via the website. The dataset will be versioned.

If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)? NA

Will older versions of the dataset continue to be supported/hosted/-maintained? We plant to continue hosting older versions of the dataset.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? Yes, we make the dataset generation code available.

B.1.8 Other questions

Is your dataset free of biases? Yes.

Can you guarantee compliance to GDPR? No, we are unable to comment on legal issues.

B.1.9 Author statement of responsibility

The authors confirm all responsibility in case of violation of rights and confirm the licence associated with the dataset and its images.

B.2 Dataset

The dataset can be accessed at https://www.robots.ox.ac.uk/~vgg/research/clevrtex. In CLEVRTEX and its variants, each instance contains:

1. RBG scene image
2. semantic mask image
3. depth mask image
4. shadow mask image
5. albedo mask image
6. normal mask image
7. Metadata JSON, which further details:
   (a) number of objects
   (b) background material
   (c) shape of each object
   (d) size of each object
   (e) rotation of each object
   (f) scene (3D) coordinates of each object
   (g) image (2D) coordinates of each object
   (h) material of each object
   (i) color (only relevant on VARBG) of each object
   (j) scene directions (CLEVR metadata)
   (k) object relationships (CLEVR metadata)

All images are provided as PNG. We also provide code for reading in the dataset and evaluation utilities for general performance metrics and per-shape/material/size breakdown. The dataset is provided under the CC-BY license.

B.3 Supplementary Material

B.3.1 Data

All images are center-cropped to a 192×192 patch and further downsampled to 128×128 pixels as a pre-processing step before being fed to the models. This introduces partially visible objects in the datasets, removes uninteresting empty edges of the scenes, and lowers the computational load. Many of the benchmarked models were developed to work with such resolution. We include helper code to load our datasets for convenience. For CLEVR we are using a version that includes segmentation masks for evaluation[1], for which we adopt the standard 70k/15k/15k train/validation/test splits.

B.3.2 Metrics

As previously mentioned, prior work [Greff et al. 2019; Engelcke et al. 2020; Locatello et al. 2020] evaluated using the adjusted Rand index (ARI) metric calculated only on pixels that correspond to the foreground objects, filtered using ground-truth data. We share the concern of some authors [Engelcke et al. 2020; Monnier et al. 2021] that such evaluation protocol does not account for whether objects are considered a part of the background and how well models segment object boundaries. Instead, we opt for the mIoU metric, familiar from the supervised segmentation setting. The predicted objects are matched with ground truth segments using the Hungarian matching algorithm, which assigns only a single predicted component to each true mask, maximizing overall overlap. A mean is taken over all objects, including the background. We provide side-by-side comparison of these metrics on all benchmarked models in tables B.1 and B.2. We chose mIoU in favor of ARI metric, as it weights all objects equally irrespective of their size. ARI is based on counting pairs, thus it gives larger regions such backgrounds more weight.

B.3.3 Hyper-parameters

Where available in PyTorch, we use the official implementation for the benchmarked methods. Otherwise, we use a re-implementation, checked against the original method, and further verify that it produces similar results to those reported in the

[1] Available at https://github.com/deepmind/multi_object_datasets.

Table B.1: Benchmark results on CLEVR, ClevrTex, CAMO, and OOD comparing ARI-FG and mIoU metrics. Results are shown ($\pm\sigma$) calculated over 3 runs.

Model	CLEVR		ClevrTex		OOD		CAMO	
	↑ARI-FG (%)	↑mIoU (%)	↑ARI-FG (%)	↑mIoU (%)	↑ARI-FG (%)	↑mIoU (%)	↑ARI-FG (%)	↑mIoU (%)
⊡ SPAIR* [Crawford and Pineau 2019]	77.13 ± 1.92	65.95 ± 4.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
⊡ SPACE [Z. Lin et al. 2020b]	22.75 ± 14.04	26.31 ± 12.93	17.53 ± 4.13	9.14 ± 3.46	12.71 ± 3.44	6.87 ± 3.32	10.55 ± 2.09	8.67 ± 3.50
⊡ GNM [Jiang and Ahn 2020]	65.05 ± 4.19	59.92 ± 3.72	53.37 ± 0.67	42.25 ± 0.18	48.43 ± 0.86	40.84 ± 0.30	15.73 ± 0.89	17.56 ± 0.74
♣ MN [Smirnov et al. 2021]	72.12 ± 0.64	56.81 ± 0.40	38.31 ± 0.70	10.46 ± 0.10	37.29 ± 1.04	12.13 ± 0.19	31.52 ± 0.87	8.79 ± 0.15
♣ DTI [Monnier et al. 2021]	89.54 ± 1.44	48.74 ± 2.17	79.90 ± 1.37	33.79 ± 1.30	73.67 ± 0.98	32.55 ± 1.08	72.90 ± 1.89	27.54 ± 1.55
⊞ GenV2 [Engelcke et al. 2021]	57.90 ± 20.38	9.48 ± 0.55	31.19 ± 12.41	7.93 ± 1.53	29.04 ± 11.23	8.74 ± 1.64	29.60 ± 12.84	7.49 ± 1.67
⊞ eMORL [Emami et al. 2021]	93.25 ± 3.24	50.19 ± 22.56	45.00 ± 7.77	12.58 ± 2.39	43.13 ± 9.28	13.17 ± 2.58	42.34 ± 7.19	11.56 ± 2.09
⊞ MONet [C. P. Burgess et al. 2019]	54.47 ± 11.41	30.66 ± 14.87	36.66 ± 0.87	19.78 ± 1.02	32.97 ± 1.00	19.30 ± 0.37	12.44 ± 0.73	10.52 ± 0.38
⊞ SA [Locatello et al. 2020]	95.89 ± 2.37	36.61 ± 24.83	62.40 ± 2.23	22.58 ± 2.07	58.45 ± 1.87	20.98 ± 1.59	57.54 ± 1.01	19.83 ± 1.41
⊞ IODINE [Greff et al. 2019]	93.81 ± 0.76	45.14 ± 17.85	59.52 ± 2.20	29.17 ± 0.75	53.20 ± 2.55	26.28 ± 0.85	36.31 ± 2.57	17.52 ± 0.75

Table B.2: Results on PlainBG,VarBG, and GrassBG variants, comparing ARI-FG and mIoU metrics.

Model	PlainBG		VarBG		GrassBG	
	↑ARI-FG (%)	↑mIoU (%)	↑ARI-FG (%)	↑mIoU (%)	↑ARI-FG (%)	↑mIoU (%)
⊡ SPAIR* [Crawford and Pineau 2019]	51.75	39.32	0.05	0.00	0.00	0.00
⊡ SPACE [Z. Lin et al. 2020b]	34.25	31.96	29.36	16.10	32.52	33.85
⊡ GNM [Jiang and Ahn 2020]	40.73	26.49	66.79	49.78	67.31	53.15
♣ MN [Smirnov et al. 2021]	38.34	10.16	43.64	11.51	59.79	34.80
♣ DTI [Monnier et al. 2021]	77.74	36.03	81.56	38.82	82.37	37.65
⊞ GenV2 [Engelcke et al. 2021]	85.33	24.39	66.04	14.40	21.12	2.88
⊞ eMORL [Emami et al. 2021]	52.00	29.39	50.18	22.92	69.64	19.38
⊞ MONet [C. P. Burgess et al. 2019]	57.10	38.72	51.87	23.73	37.97	21.29
⊞ SA [Locatello et al. 2020]	51.75	39.32	89.78	62.57	43.55	12.88
⊞ IODINE [Greff et al. 2019]	54.32	23.83	75.33	39.86	66.91	25.76

corresponding papers. Where the original methods have been applied to CLEVR (or its variant), we employ the same hyper-parameter configuration for CLEVR. For other datasets or methods that have not been trained on CLEVR, we follow a best-effort approach to tuning hyper-parameters.

For MONet [C. P. Burgess et al. 2019], we reduced the batch size from 64 to 63 (3×21). IODINE [Greff et al. 2019] and MONet were trained for 300k iterations instead of 1M as we noticed that no changes to learned configurations, running loss, or performance improvements were taking place after 250k iterations. For MONet, IODINE we found the original configuration worked well enough. For SPACE [Z. Lin et al. 2020b], we concentrated on finding a suitable setting for output standard deviation for foreground and background networks. Despite higher values being crucial for both Genesis and GNM models, we could not identify a configuration that produced better results than the original 0.15 in our exploration. The following describes any adjustments made to the original configurations for other models.

Slot Attention [Locatello et al. 2020] We use 11 slots on all tests. We varied the number of attention iterations. We have found the model to perform the best when trained using 3. We maintained the original learning rate, batch size, and

optimizer settings and trained for the suggested 500k iterations.

Efficient MORL [Emami et al. 2021] We increase the number of components to 11 and change the input resolution to 128×128 to be inline with other methods studied. GECO reconstruction target is further adjusted to account for change in resolution. We use the value of $-108\,000$ for CLEVR and PlainBG. We use higher values of $-61\,000$ for VarBG and GrassBG and $-73\,000$ for ClevrTex, OOD, and CAMO, due to more complex backgrounds. We considered a set of $\{-8\,000, -48\,000, -61\,000, -69\,000, -73\,000, -108\,000, -112\,000\}$, selecting the best performing ones. eMORL†: following the release of ClevrTex, the codebase of eMORL has been updated including configuration settings for ClevrTex. The authors provided us with trained models that show better performance (table 2.3) in our evaluation.

GNM [Jiang and Ahn 2020] We use a 4×4 slot grid with total of 16 slots and a latent dimension of 64 for objects and 10 for background. We found the model extremely sensitive to the output standard deviation. We found values 0.2 on CLEVR and 0.5 on ClevrTex worked well. It is worth noting that in our testing, with values of 0.4 and 0.6, GNM could not learn to segment the scene. We trained for 300k iteration.

GenesisV2 [Engelcke et al. 2021] We focused our hyper-parameter selection on the output standard deviation and GECO [Rezende and Viola 2018] objective. On CLEVR we used GECO goal of 0.5655 and output standard deviation of 0.7, which was crucial for model to learn as lower values did not produce good segmentations. On ClevrTex we lowered the GECO goal to 0.5, which outperformed CLEVR setting.

SPAIR* [Crawford and Pineau 2019] As mentioned before, we incorporated a background VAE network into SPAIR by using a convolutional encoder and a spatial broadcast decoder [Watters et al. 2019b]. We also replaced MLP-based glimpse decoder with a similar spatial broadcast decoder. Additionally, we added an extra convolution in the backbone network to handle inputs of 128×128 size. In this configuration, SPAIR had 16 slots. We set the latent dimension of objects to 64, and background to 1 on CLEVR and 4 on ClevrTex. We trained for 250k

iterations using a batch size of 128, Adam optimizer, learning rate of 1e-4, with gradient clipping when norm exceeded 1.0. We used β value of 2.7. On CLEVR we used the output standard deviation of 0.15. On CLEVRTEX, we annealed the value from 0.5 to 0.15 over 50k iterations. On CLEVR, the object presence prior hyper-parameter s was annealed from 0.0001 to 0.99 over 10k, on CLEVRTEX, over 50k iterations.

DTISprites [Monnier et al. 2021] On CLEVR, we used the setting used for CLEVR6 in the original work except for increasing the possible number of objects. We found that using ten slots leads to better segmentation results than setting to 11 as with other models (one more than the max number of objects). On CLEVRTEX, we used color and protective transforms for both sprites and backgrounds.

Table B.3: Architecture of component networks changed in SPAIR*.

Conv Encoder

Layer	Size/Ch.	Act.	Comment
Conv 3×3	32	ReLU	stride 2
Conv 3×3	32	ReLU	stride 2
Conv 3×3	64	ReLU	stride 2
Conv 3×3	64	ReLU	stride 2
Avg P 1×1			
MLP	128	ReLU	
MLP	$\|\|\mu\|\| + \|\|\sigma\|\|$	Softplus for σ only	

Broadcast Decoder

Layer	Size/Ch.	Act.	Comment
Broadcast			add coord.
Conv 3×3	32	ReLU	no pad
Conv 3×3	32	ReLU	no pad
Conv 3×3	32	ReLU	no pad
Conv 3×3	32	ReLU	no pad
Conv 1×1	4	Sigmoid for masks only	

MarioNette [Smirnov et al. 2021] We adjusted the model to learn to select and use from a dictionary of backgrounds, same as sprites. Additionally, we lowered the layer size to 4, using two layers, which gives 32 possible slots of size 64×64. On CLEVRTEX, we increased the sizes of both background and sprite dictionaries to as large as would fit in GPU memory. We trained with 60 sprites and single background on CLEVR, PLAINBG, and GRASSBG increasing the number of backgrounds to 60 on VARBG and CLEVRTEX.

B.3.4 Extra Figures

Here, we include extra figures listing additional output for all benchmarked models on CLEVR, CLEVRTEX, test sets and variants (fig. B.1). fig. B.2 contains example output of sprite- and glimpse-based models when they fail to learn correct foreground and background elements and learn to tile the image instead.

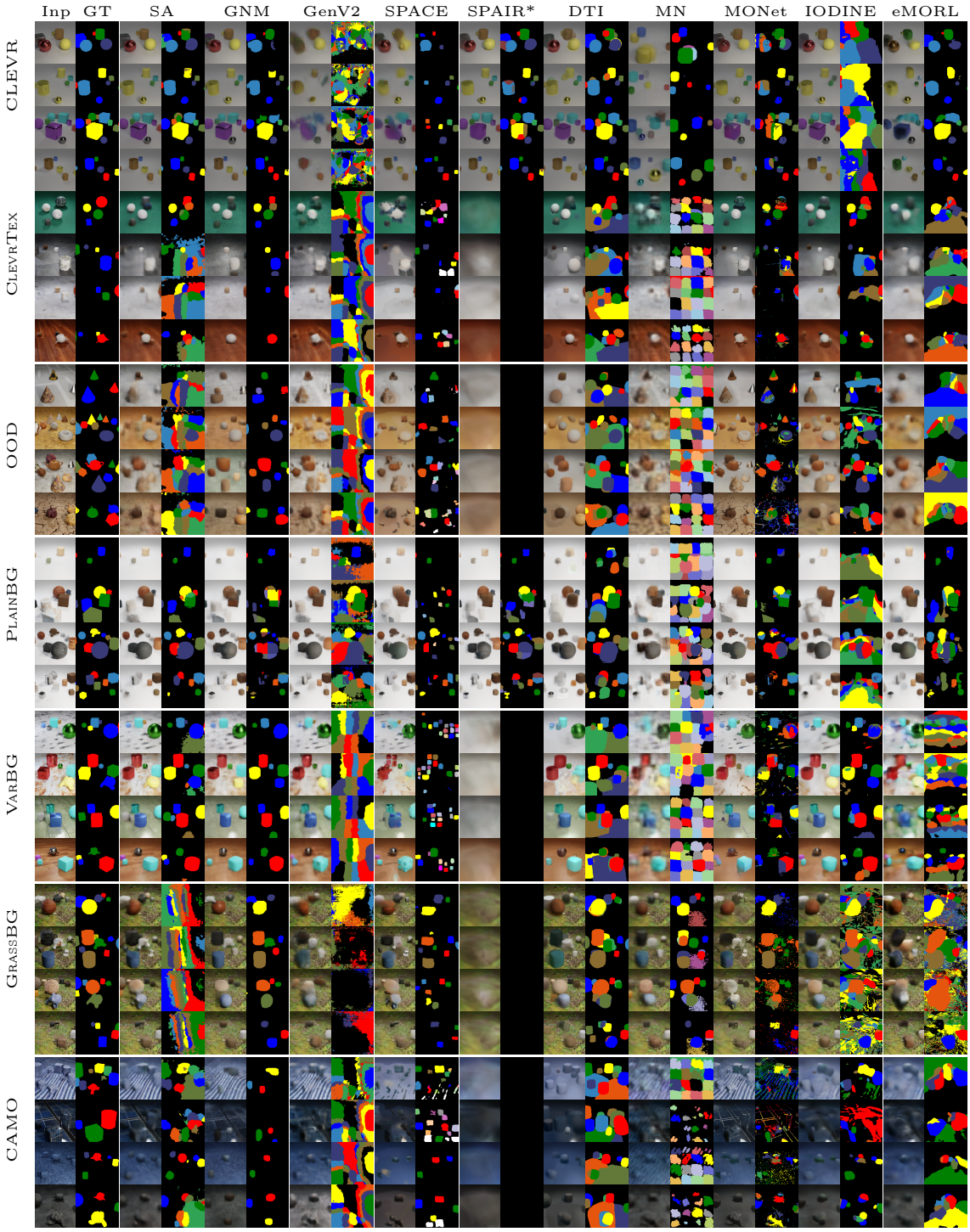


Figure B.1: Comparison of various model reconstruction and segmentation outputs on CLEVR, CLEVRTEX and variants. Best viewed digitally.

B.3.5 Dataset Construction

The main method of how the dataset is constructed is described in section 2.3.1. Here, we include additional figures to showcase some steps in the dataset creation and provide catalog of materials used.

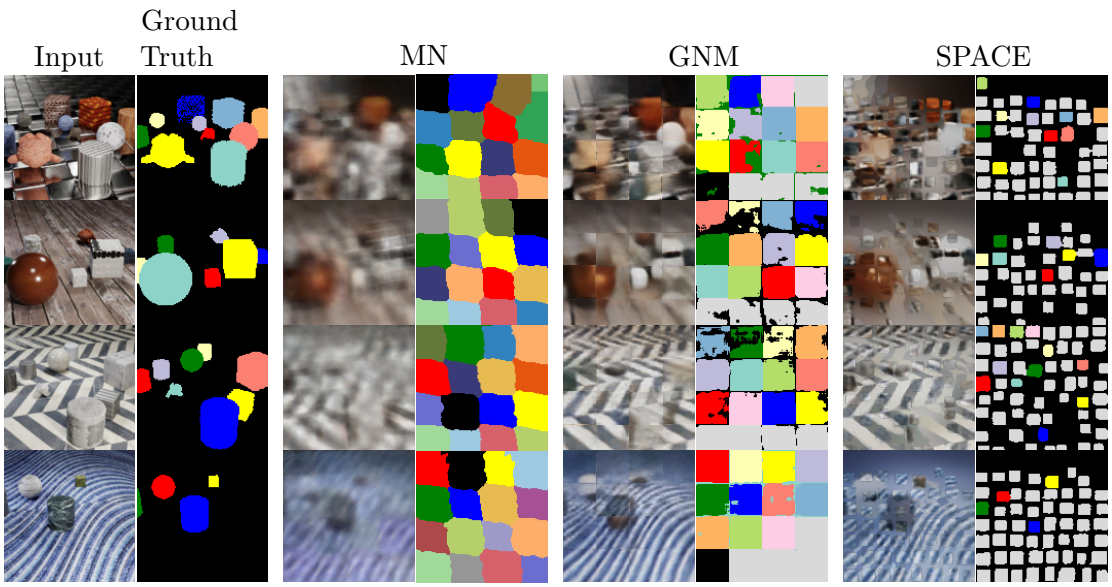


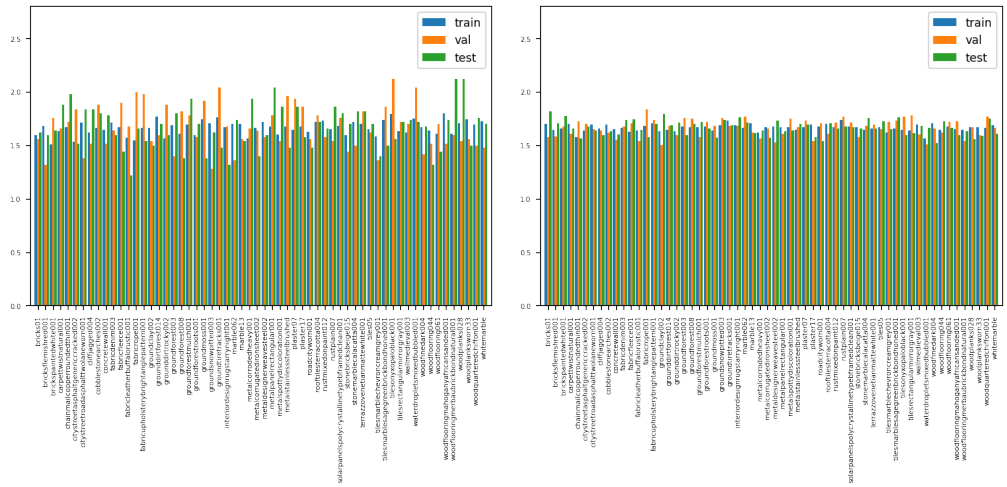
Figure B.2: Tiling behaviour common to glimpse- (⊡) and sprite-based (♣) models. Such tiling occurred whenever the model could not reproduce the foreground and background elements with respective component networks to sufficient accuracy. The models are trained on CLEVRTEX. GNM is shown here trained with output $\sigma = 0.3$.

Lighting fig. B.4 shows the possible range of randomizing light positions in the scene, from warm closeup light positions with lots of shadows falling onto other objects to distant lights casting small soft shadows onto background even in crowded scenes. fig. B.4 also shows 4 possible shapes at 3 possible scales used in the CLEVRTEX.

Shape Adjustments CLEVRTEX features only 4 simple objects. This is mitigated by a range of material-specific geometry adjustments, bumping and transparency mapping applied to the seed shapes. fig. B.5 shows the effect of the shape perturbations in a scene where no other material properties have been applied to the objects.

Camera The camera position is jitterred along with lights. We use a perspective camera with a focal length of 0.035m and 0 shift.

Dataset Splits CLEVRTEX and variants are split into test/val/train datasets using 10%/10%/80% proportions after generation. The splits are made based on the index of the example, that is first 10% form test split. This simple scheme is motivated by the uniform sampling of the scene composition. fig. B.3 shows that



(a) Background materials (b) Object materials

Figure B.3: Distribution of 60 materials in CLEVRTEX dataset between train/-val/test splits, shown as a percentage. (a) shows distribution for the background. (b) shows distribution for objects.

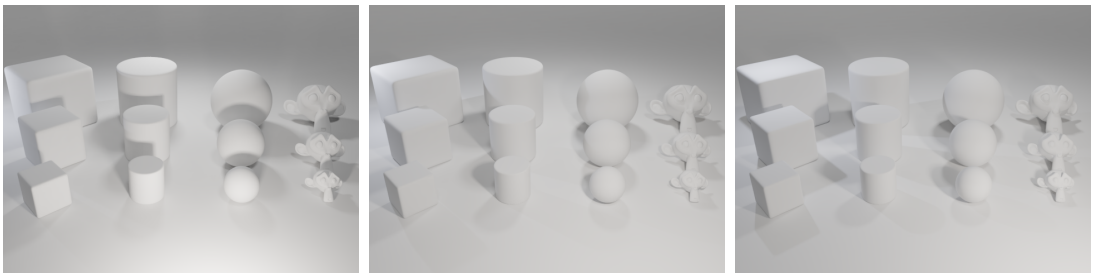


Figure B.4: Effects of jittering light positions in the scenes. The images show two extremes with the mean position in the middle. The images also contain a showcase of 4 shapes present in the main CLEVRTEX dataset at 3 possible scales. The scenes are rendered without any materials.

this results in roughly proportional distribution of materials for both backgrounds and objects across dataset splits. OOD variant is test-only.

Materials fig. B.6 contains the list of 60 materials used in generating CLEVRTEX and its PLAINBG, VARBG, GRASSBG, and CAMO variants. Please see our generation code for further information. fig. B.7 contains 25 materials used in OOD variant.

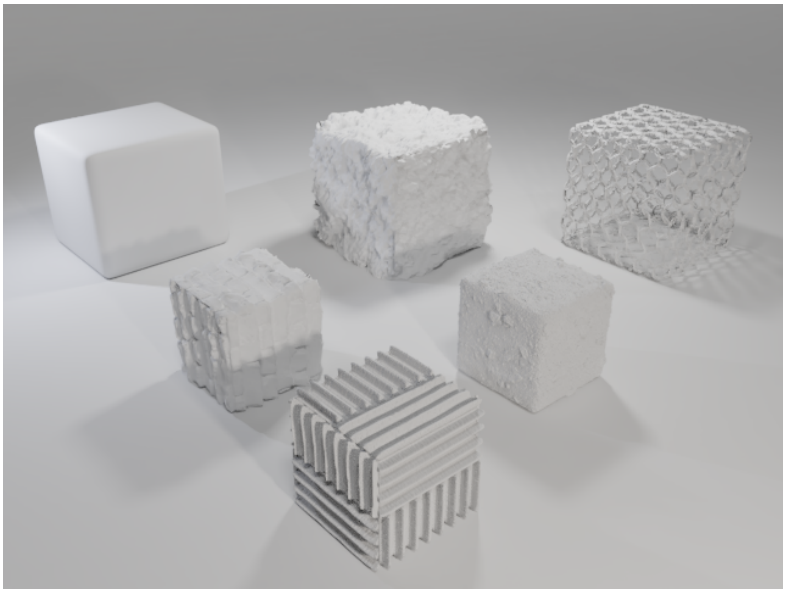


Figure B.5: Showcase of a diverse set of shape perturbations applied the basic cube (top left) through a combination of displacement mapping, bumping and transparency mapping. Other material properties are not applied to the objects to show only the displacement details.

Figure B.6: Materials used in CLEVRTEX dataset.

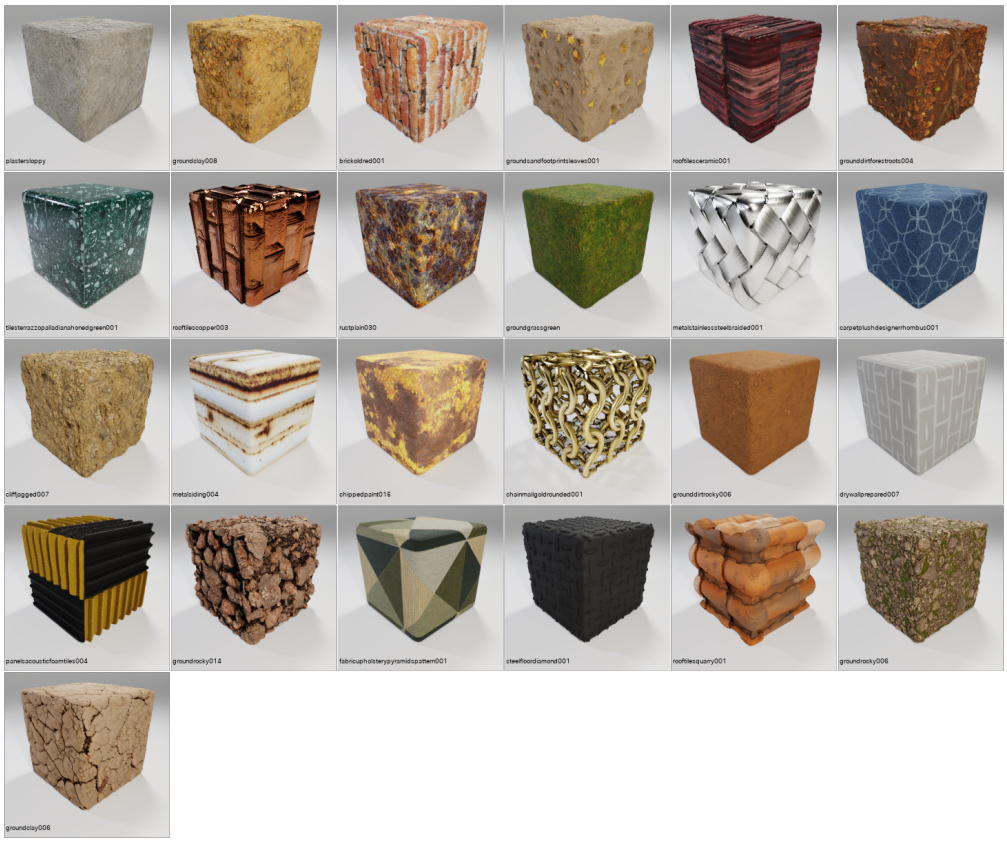


Figure B.7: Materials used in OOD dataset variant.

Appendix C

Guess What Moves: Unsupervised Video and Image Segmentation by Anticipating Motion Supplementary Material

In this supplementary material, we provide further details on our training parameters in Appendix C.1. Appendix C.2 contains the closed form solution of the fitting of the flow model θ. Expanded experiments and ablations are found in Appendix C.3. Finally, more qualitative results are presented in appendix C.4. See the project page, https://www.robots.ox.ac.uk/~vgg/research/gwm, for additional visualizations, code and models.

C.1 Experimental Setup

Network. We use MaskFormer [B. Cheng et al. 2021] as our segmentation network[1], and use only the segmentation head. As MaskFormer predicts masks at 4 times lower resolution than input, we modify the `PixelDecoder` by appending [*Conv*(3), *UpsampleNN(2)*, *Conv*(1)]×2 to its output layers to bring the masks back up to the input resolution.

For the backbone and appearance features V, we leverage a ViT-8 transformer,

[1]Implementation from https://github.com/facebookresearch/MaskFormer.

pre-trained on ImageNet [Russakovsky et al. 2015] in a self-supervised manner using DINO [Caron et al. 2021] to avoid any external sources of supervision. For the hierarchical backbone features to decoder we use the key feature outputs from layers 6, 8, 10, 12.

The input RGB images are interpolated (bi-cubic) to 128×224 resolution for input to the network. We interpolate (nearest neighbor) the optical flow to 480×854 for the loss. Output segmentation logits are up-sampled using bi-linear interpolation to the flow resolution for training and again to annotation resolution for evaluation.

Training Hyperparameters. The networks are optimised using AdamW [Loshchilov and Hutter 2018], with learning rate of 1.5×10^{-4}, a schedule of linear warm-up from 1.0×10^{-6} to 1.5×10^{-4} over 1.5k iteration and polynomial decay afterwards. We use batch size of 8 and train for 15k iterations. We additionally employ gradient clipping when the 2-norm exceeds 0.01 for stability. The loss multiplier is 0.03.

UNet. For experiments using U-Net[2], we use the standard 4-layer version. The batch-size is increased to 16 and learning rate to 7.0×10^{-4}. We also clip the gradients only when 2-norm exceeds 5.0. All other settings, including optimizer and learning rate schedules, are kept the same. U-Net is not pre-trained and trained from scratch.

Optical Flow. Our method derives its learning signal from optical flow estimated using off-the-shelf frozen networks. We estimate optical flow for all frames on DAVIS, STv2, and FBMS following the practice of MotionGrouping [Charig Yang et al. 2021]. We employ RAFT [Teed and J. Deng 2020] (supervised) using the original resolution for our main experiments, and gaps between frames of $\{-2, -1, 1, 2\}$ for DAVIS and STv2, and $\{-6, -3, 3, 6\}$ on FBMS. When multiple flows are associated with a single frame (multiple gaps), we sample one at random for each iteration.

[2]Implementation from https://github.com/milesial/Pytorch-UNet.

C.2 Quadratic Flow Model: Closed Form Solution

Consider one of K regions m and define $w_u \propto P(m_u = k|I, \Phi)$ the posterior probability for that region, normalized so that $\sum_{u\in\Omega} w_u = 1$ (the scaling factor does not matter for the purpose of finding the minimizer). We can obtain the minimizer (A^*, b^*) and minimum of the energy

$$E(A, b) = \sum_{u\in\Omega} w_u \|F_u - Au - b\|^2 \tag{C.1}$$

as follows. Defining

$$\bar{u} := \begin{bmatrix} u \\ 1 \end{bmatrix}, \quad M := \begin{bmatrix} A & b \end{bmatrix} \in \mathbb{R}^{2\times 6}$$

allows rewriting the energy as

$$E(M) = \sum_{u\in\Omega} w_u \|F_u - M\bar{u}\|^2 = \operatorname{tr}\left(\Lambda_{FF} - M\Lambda_{\bar{\Omega}F} - \Lambda_{F\bar{\Omega}}M^\top + M\Lambda_{\bar{\Omega}\bar{\Omega}}M^\top\right),$$

where

$$\Lambda_{FF} = \sum_{u\in\Omega} w_u F_u F_u^\top, \quad \Lambda_{F\bar{\Omega}} = \sum_{u\in\Omega} w_u F_u \bar{u}^\top, \quad \Lambda_{\bar{\Omega}F} = \Lambda_{F\bar{\Omega}}^\top, \quad \Lambda_{\bar{\Omega}\bar{\Omega}} = \sum_{u\in\Omega} w_u \bar{u}\bar{u}^\top.$$

are the (uncentered) second moment matrices of the flow F_u and homogeneous coordinate vectors $\bar{u}$. By inspection of the trace term, the gradient of the energy is given by:

$$\frac{dE(M)}{dM} = 2\left(\Lambda_{F\bar{\Omega}} - M\Lambda_{\bar{\Omega}\bar{\Omega}}\right)$$

Hence, the optimal regression matrix M^* and corresponding energy value are

$$M^* = \Lambda_{F\bar{\Omega}}\Lambda_{\bar{\Omega}\bar{\Omega}}^{-1}, \qquad E(M^*) = \operatorname{tr}\left(\Lambda_{FF} - M^*\Lambda_{\Omega\bar{F}}\right).$$

Somewhat more intuitive results can be obtained by centering the moments and

resolving for A and b instead of M. Specifically, define:

$$\mu_\Omega := \sum_{u\in\Omega} w_u u, \quad \mu_F := \sum_{u\in\Omega} w_u F_u.$$

The covariance matrices of the vectors are:

$$\Sigma_{FF} = \sum_{u\in\Omega} w_u (F_u - \mu_F)(F_u - \mu_F)^\top, \quad \Sigma_{F\Omega} = \sum_{u\in\Omega} w_u (F_u - \mu_F)(u - \mu_\Omega)^\top,$$
$$\Sigma_{\Omega F} = \Lambda_{F\Omega}^\top, \quad \Sigma_{\Omega\Omega} = \sum_{u\in\Omega} w_u (u - \mu_\Omega)(u - \mu_\Omega)^\top.$$

It is easy to check that

$$\Lambda_{FF} = \Sigma_{FF} + \mu_F \mu_F^\top, \quad \Lambda_{F\bar{\Omega}} = \begin{bmatrix} \Sigma_{F\Omega} + \mu_F \mu_\Omega^\top & \mu_F \end{bmatrix}, \quad \Lambda_{\bar{\Omega}\bar{\Omega}} = \begin{bmatrix} \Sigma_{\Omega\Omega} + \mu_\Omega \mu_\Omega^\top & \mu_\Omega \\ \mu_\Omega^\top & 1 \end{bmatrix}.$$

From this:

$$\begin{aligned} M^* = \Lambda_{F\bar{\Omega}} \Lambda_{\bar{\Omega}\bar{\Omega}}^{-1} &= \begin{bmatrix} \Sigma_{F\Omega} + \mu_F \mu_\Omega^\top & \mu_F \end{bmatrix} \begin{bmatrix} \Sigma_{\Omega\Omega} + \mu_\Omega \mu_\Omega^\top & \mu_\Omega \\ \mu_\Omega^\top & 1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \Sigma_{F\Omega} + \mu_F \mu_\Omega^\top & \mu_F \end{bmatrix} \begin{bmatrix} \Sigma_{\Omega\Omega}^{-1} & -\Sigma_{\Omega\Omega}^{-1}\mu_\Omega \\ -\mu_\Omega^\top \Sigma_{\Omega\Omega}^{-1} & 1 + \mu_\Omega^\top \Sigma_{\Omega\Omega}^{-1} \mu_\Omega \end{bmatrix} \\ &= \begin{bmatrix} \Sigma_{F\Omega}\Sigma_{\Omega\Omega}^{-1} & \mu_F - \Sigma_{F\Omega}\Sigma_{\Omega\Omega}^{-1}\,\mu_\Omega \end{bmatrix} = \begin{bmatrix} A^* & b^* \end{bmatrix}. \end{aligned}$$

Hence, the optimal regression coefficients and energy value are also given by:

$$A^* = \Sigma_{F\Omega}\Sigma_{\Omega\Omega}^{-1}, \quad b^* = \mu_F - A^*\mu_\Omega.$$

C.3 Further Experiments

C.3.1 Generalization in Unsupervised Video Segmentation

We also test our model in a video *generalization* setting. In contrast to the protocol of [Yanchao Yang et al. 2019; Charig Yang et al. 2021], where evaluation set is observed together with training to infer masks jointly[3], here we train only on frames from the training set. We report performance on unseen videos. In this case, our

[3]Note, no annotations are observed at any point.

	Model	Flow	**DAVIS** ($\mathcal{J}\uparrow$)	**FBMS** ($\mathcal{J}\uparrow$)
[R. Liu et al. 2021]	AMD (100 steps)	✗	57.8	47.5
	Ours (Zero shot)	ARFlow	62.5	65.4
	Ours (20 steps)	ARFlow	65.2	67.6
[Meunier et al. 2022]	EM	RAFT	69.3	57.8
	Ours (Zero shot)	RAFT	66.8	73.2
	Ours (20 steps)	RAFT	76.3	77.1

Table C.1: **Generalization performance on unseen *videos*.** Few unsupervised methods operate in this setting. AMD trains on YT-VOS, followed by 100 test-time adaptation steps, while EM trains on FlyingThings3D using flow as input. We use (fully unsupervised) ARFlow for fair comparison with AMD. Our method shows better performance after observing motion. (Test-time adaptation uses the training loss. No GT is involved at any point.)

Backbone model	Backbone pretraining	Sup.	**DAVIS** $\mathcal{J}\uparrow$	**STv2** $\mathcal{J}\uparrow$	**FBMS** $\mathcal{J}\uparrow$
ViT-8	ImageNet DINO	✗	79.5	78.3	77.4
UNet	None	✗	78.3	76.8	72.0
SWIN-tiny	ImageNet MOBY	✗	78.3	77.4	74.6
SWIN-tiny	ImageNet CLS	✓	78.9	77.7	75.5
SWIN-tiny	None	✗	78.3	75.2	68.8
Resnet-50	ImageNet CLS	✓	77.5	75.8	72.9

Table C.2: **Effect of Pretraining/Backbone.** Our method with MaskFormer benefits from pretraining, with slight improvement offered by supervised (*CLS*) over unsupervised (*MOBY*) pretraining (usng SWIN transformer). Comparable results can be obtained with training from scratch. Best results are obtained using DINO features.

method independently segments a collection of frames from a new video, with no way to incorporate motion information.

To "observe" motion on *unseen* inputs, we also report results after taking 20 test-time adaptation steps (using our unsupervised loss) for each evaluation sequence in isolation (c.f. AMD [R. Liu et al. 2021] takes 100 test-time adaptations steps). That is after training, we follow our training setup (optimizer, rate, batch size) and feed frames from the evaluation video and corresponding optical flow, calculate loss and take gradient steps. Despite other methods using much larger training sets, our approach shows better performance (table C.1).

C.3.2 Ablation Studies

Pretraining. Compared to recent methods for video segmentation [Charig Yang et al. 2021; Meunier et al. 2022], one of the benefits of our formulation is that we

Model	K	Merge	DAVIS $\mathcal{J}\uparrow$	STv2 $\mathcal{J}\uparrow$	FBMS $\mathcal{J}\uparrow$
Ours	$K=4$	✓	**79.5**	**78.3**	**77.4**
Spectral clustering	$K=2$	✗	15.79	14.89	27.45
K-Means	$K=4$	✓	41.79	34.84	48.80
K-Means	$K=2$	✗	20.24	21.14	38.25

Table C.3: **Feature Clustering without Motion.** We experiment with offline clustering of DINO features to assess the importance of our motion-based formulation. Simply clustering DINO features using K-Means or spectral clustering [Melas-Kyriazi et al. 2022a] into 2 clusters performs worse. Over-clustering and merging using our cluster-merging approach performs better but still fails to reach our performance.

can leverage unsupervised pretraining for the segmentation network (e.g., for the ViT backbone of MarkFormer). This enables our method to be trained in only 15k iterations. Here, we investigate the importance of the backbone. To this end we replace ViT with Swin-tiny pretrained using MOBY (self-supervised) in table C.2. The performance differences are small.

Additionally, we investigate the effect of other pretraining strategies on the performance. Switching to a model pretrained on ImageNet with image-level supervision (i.e. a classification task) only slightly improves performance showing that the method does not need to rely on supervised pre-training. Finally, we train the model using same settings for 20k iterations from scratch, without any pre-training. This results in comparable performance on DAVIS but reduced performance on the smaller datasets. Comparing backbones without pre-training, UNet gives better results than SWIN-tiny, likely due to smaller networks being easier to train on small datasets.

Feature Clustering without Motion. To demonstrate the potential of using motion for discovering objects, in table C.3, we compare to additional baselines that only rely on clustering visual features. Spectral feature clustering with $K=2$ (based on [Melas-Kyriazi et al. 2022a]), on the same visual features we use to merge segments (i.e., DINO) after over-clustering, shows (somewhat unsurprisingly) that learning from motion is important for motion segmentation. Similarly, K-means ($K=2$) on the same features also falls behind our method. Yet, we show that K-means also benefits from over-clustering ($K=4$) and then merging.

Opt. Flow		Sup.	**DAVIS** ($\mathcal{J}\uparrow$)
[Liang Liu et al. 2020]	ARFlow	✗	66.9
[D. Sun et al. 2018]	PWCNet	✓	74.9
[Teed and J. Deng 2020]	RAFT	✓	79.5

Table C.4: **Choice of Optical Flow Method.** Measuring the influence of the method to extract optical flow.

Method		**DAVIS** ($\mathcal{J}\uparrow$)
[Charig Yang et al. 2021]	MG	53.2
[R. Liu et al. 2021]	AMD	57.8
	Ours	**66.9**

Table C.5: **Fully Unsupervised Video Object Segmentation.** Comparison to the state of the art in unsupervised VOS without reliance on *any* supervision

		CUB			**DUTS**			**ECSSD**			**OMRON**		
		Acc	$\mathcal{J}\uparrow$	$maxF_\beta\uparrow$	Acc	$\mathcal{J}\uparrow$	$F_\beta\uparrow$	Acc	$\mathcal{J}\uparrow$	$F_\beta\uparrow$	Acc	$\mathcal{J}\uparrow$	$F_\beta\uparrow$
[X. Xia and Kulis 2017]	WNet†	–	24.8	–	–	–	–	–	–	–	–	–	–
[Ji et al. 2019a]	IIC-seg	–	36.5	–	–	–	–	–	–	–	–	–	–
[Bielski and Favaro 2019]	PertGAN	–	38.0	–	–	–	–	–	–	–	–	–	–
[M. Chen et al. 2019]	ReDO	84.5	42.6	–	–	–	–	–	–	–	–	–	–
[Kanezaki 2018]	UISB	–	44.2	–	–	–	–	–	–	–	–	–	–
[Benny and Wolf 2020]	OneGAN	–	55.5	–	–	–	–	–	–	–	–	–	–
[Yu et al. 2021]	DRC	–	56.4	–	–	–	–	–	–	–	–	–	–
[X. He et al. 2022]	GANSeg	–	62.9	–	–	–	–	–	–	–	–	–	–
[Voynov et al. 2021]	Voynov *et al.*	94.0	71.0	80.7	88.1	51.1	60.0	90.6	68.4	79.0	86.0	46.4	53.3
[R. Liu et al. 2021]	AMD	–	–	–	–	–	60.2	–	–	–	–	–	–
[Melas-Kyriazi et al. 2022b]	Kyriazi *et al.*	92.1	66.4	78.3	89.3	52.8	61.4	91.5	71.3	80.6	88.3	50.9	58.3
[Melas-Kyriazi et al. 2022a]	Kyriazi *et al.*	–	76.9	–	–	51.4	–	–	73.3	–	56.7	–	
[Yanchao Yang et al. 2021]	DyStaB†	–	–	–	–	–	–	–	–	88.1	–	–	73.9
[Y. Wang et al. 2022]	TokenCut	–	–	–	90.3	57.6	–	91.8	71.2	–	88.0	53.3	–
[Shin et al. 2022a]	SelfMask	–	–	–	92.3	62.6	–	94.4	78.1	–	90.1	58.2	–
Ours		93.5	64.6	80.9	91.5	49.2	65.6	88.5	56.1	74.3	89.3	41.31	56.3

Table C.6: **Expanded unsupervised object segmentation** benchmark CUB and three saliency detection benchmarks: DUTS, ECSSD, and DUT-OMRON (*OMRON*). † DyStaB uses CRF post-processing, supervised pre-training, and self-training on each dataset.

Flow Estimation. Finally, our method relies on optical flow estimated by frozen, off-the-shelf networks. So far we have been using RAFT [Teed and J. Deng 2020], as such optical flow network was adopted in our baselines. In table C.4, we also consider PWCNet [D. Sun et al. 2018] and fully-unsupervised ARFlow [Liang Liu et al. 2020]. We observe that the performance of the flow estimator has an impact on the final performance of our method. Finally, we compare our *fully* unsupervised model (which uses self-supervised pretraining and flow) to fully unsupervised state-of-the-art methods. Appearance-Motion Decomposition (AMD) [R. Liu et al. 2021] works end-to-end and directly extracts motion features from pairs of images with a PWCNet-like architecture, while MotionGrouping (MG) [Charig Yang et al. 2021] and our method use ARFlow [Liang Liu et al. 2020] for optical flow estimation. In table C.5 we show that our method achieves a significant improvement over previous approaches.

C.4 Additional Results and Discussion

We provide a further breakdown of our results in tables C.7 to C.9, reporting per sequence evaluation results on the video segmentation tasks.

Video object segmentation and egomotion. We note that some sequences have pronounced egomotion (e.g., camera shaking in `libby` of DAVIS or inside a moving car in `camel01` of FBMS). Our model performs well on these sequences, demonstrating that it can handle egomotion. When *only* the camera is moving, the resulting optical flow would still highlight objects due to parallax. This provides a learning signal, however, it would likely be weaker for objects farther away from the camera. As our method works on a per-frame basis and does not *require* flow during inference, this should not have an impact at test time. However, fine-tuning on scenes with only egomotion (see appendix C.3.1 for experiments investigating test-time adaptation) and only small or far away objects, might lead to the model learning to ignore them.

Image segmentation. For unsupervised image segmentation, we show some additional qualitative results for CUB in fig. C.1, DUT-OMRON in fig. C.2, DUTS in fig. C.3, and ECSSD in fig. C.4. Our model, trained on a combined dataset of DAVIS, FBMS and STv2, is robust enough to handle a wide array of classes from the above datasets in varying context. Our model can segment both stationary and non-stationary objects and works well when multiple objects are in the foreground. In fig. C.5, we show a few failure cases for all datasets, where the model struggles mostly with ambiguous foreground objects and, in particular, with close-ups of stationary objects, e.g. signs (ECSSD) and buildings (DUT-OMRON). The model also has issues with boundaries for many objects, i.e. the foreground objects are correctly identified but the model fails to fully segment them. For example, in DUTS, the snake in the first image has a well segmented head, however, the model does not segment its body accurately.

Sequence	*w/o CRF*			*w/ CRF*		
	$\mathcal{J}$(M)	$\mathcal{J}$(R)	$\mathcal{J}$(D)	$\mathcal{J}$(M)	$\mathcal{J}$(R)	$\mathcal{J}$(D)
blackswan	67.0	100.0	-0.8	67.4	100.0	1.1
bmx-trees	58.2	76.9	19.9	59.8	76.9	17.5
breakdance	86.2	100.0	4.9	87.4	100.0	5.2
camel	89.4	100.0	5.7	90.6	100.0	5.5
car-roundabout	81.4	90.4	26.7	81.2	90.4	25.8
car-shadow	84.3	100.0	9.0	83.9	100.0	8.0
cows	90.4	100.0	3.4	91.3	100.0	3.2
dance-twirl	87.4	100.0	-7.1	88.8	100.0	-6.2
dog	92.9	100.0	-1.7	93.9	100.0	-1.6
drift-chicane	78.6	98.0	2.2	82.0	100.0	2.6
drift-straight	80.6	100.0	7.2	82.1	100.0	8.2
goat	78.6	100.0	1.7	75.8	100.0	4.5
horsejump-high	84.9	100.0	6.4	88.0	100.0	4.6
kite-surf	64.4	97.9	4.5	67.5	97.9	3.1
libby	82.9	100.0	8.6	84.5	100.0	8.6
motocross-jump	74.1	78.9	4.1	75.1	81.6	4.1
paragliding-launch	62.2	65.4	33.5	64.1	66.7	35.8
parkour	86.1	100.0	-4.5	88.1	100.0	-3.1
scooter-black	82.1	97.6	-4.3	82.1	100.0	-4.3
soapbox	79.2	100.0	-2.8	81.0	100.0	-0.4
Average	79.5	95.3	5.8	80.7	95.7	6.1

Table C.7: **Result breakdown on DAVIS16 validation sequences.** (M), (R), and (D) are mean, recall and decay of IoU, respectively

Sequence	*w/o CRF* $\mathcal{J}$(M)	*w/ CRF* $\mathcal{J}$(M)
drift	86.1	86.5
birdfall	67.8	57.1
girl	84.5	86.3
cheetah	57.0	50.8
worm	83.7	84.0
parachute	90.6	93.2
monkeydog	22.9	22.6
hummingbird	57.3	57.2
soldier	77.4	77.4
bmx	76.4	77.5
frog	84.1	86.7
penguin	77.7	76.8
monkey	75.0	75.8
bird of paradise	92.3	94.0
Seq. Avg.	73.8	73.3
Frame Avg.	78.3	78.9

Table C.8: Sequence breakdown on SegTrackv2 dataset.

Sequence	*w/o CRF* $\mathcal{J}$(M)	*w/ CRF* $\mathcal{J}$(M)
camel01	86.8	91.0
cars1	86.9	86.8
cars10	64.6	64.8
cars4	81.5	82.4
cars5	81.6	82.1
cats01	87.7	89.5
cats03	69.4	63.2
cats06	66.5	67.4
dogs01	76.3	75.6
dogs02	85.3	86.4
farm01	90.8	90.5
giraffes01	82.1	83.9
goats01	79.9	83.7
horses02	80.4	83.6
horses04	59.8	60.5
horses05	72.8	74.5
lion01	75.1	75.0
marple12	81.9	81.6
marple2	84.4	85.9
marple4	81.1	82.4
marple6	95.1	95.1
marple7	76.6	77.6
marple9	95.4	96.3
people03	90.1	91.0
people1	85.3	87.2
people2	88.1	89.7
rabbits02	91.2	91.2
rabbits03	81.5	84.4
rabbits04	43.8	44.1
tennis	73.3	74.2
Seq. Avg.	79.8	80.7
Frame Avg.	77.4	78.4

Table C.9: Sequence breakdown on FBMS59 dataset

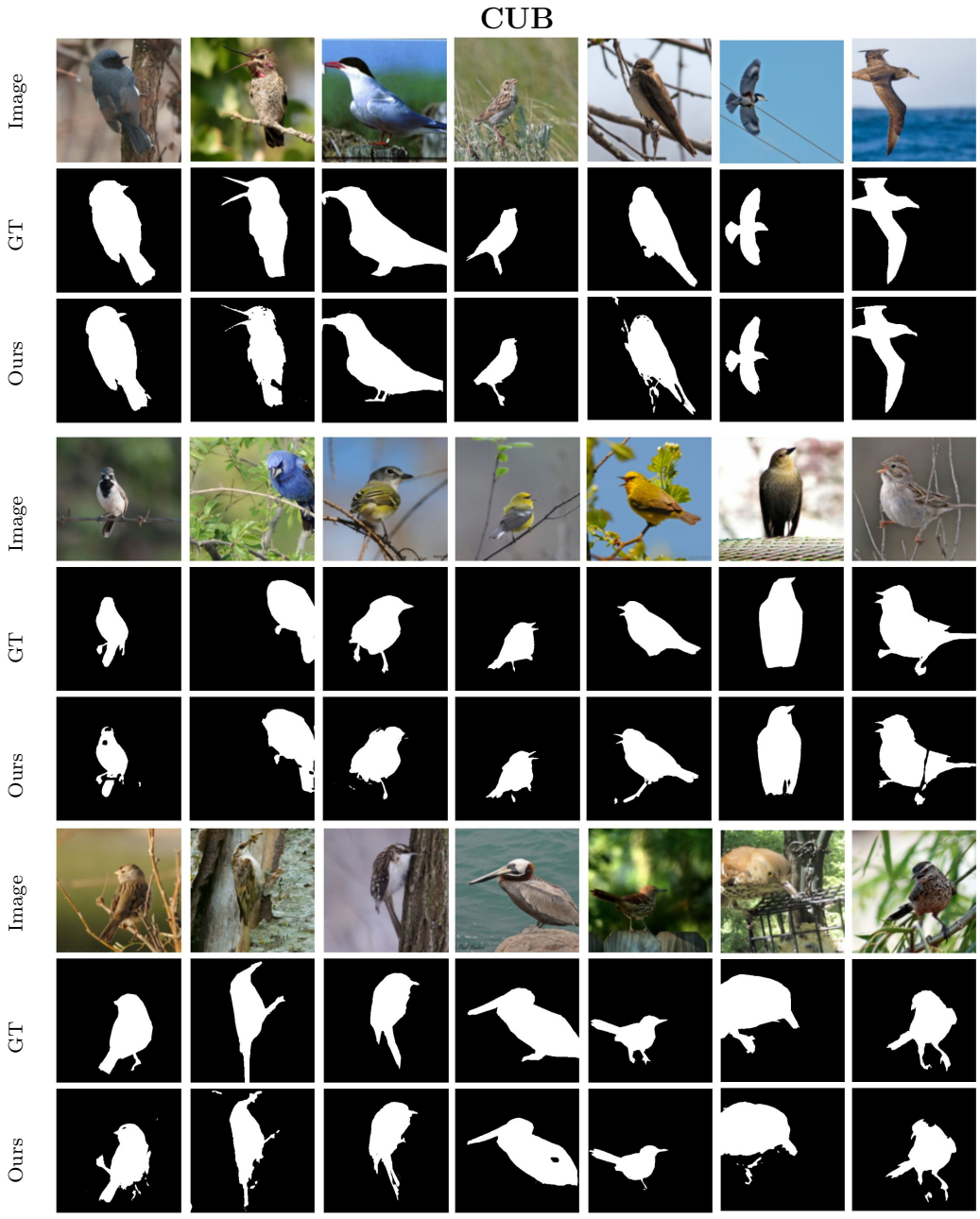


Figure C.1: **Qualitative Comparison on CUB**. We train our model on a combined dataset of DAVIS, FBMS and STv2. Our method can extract birds in different environments and poses. Our model can segment different species of birds

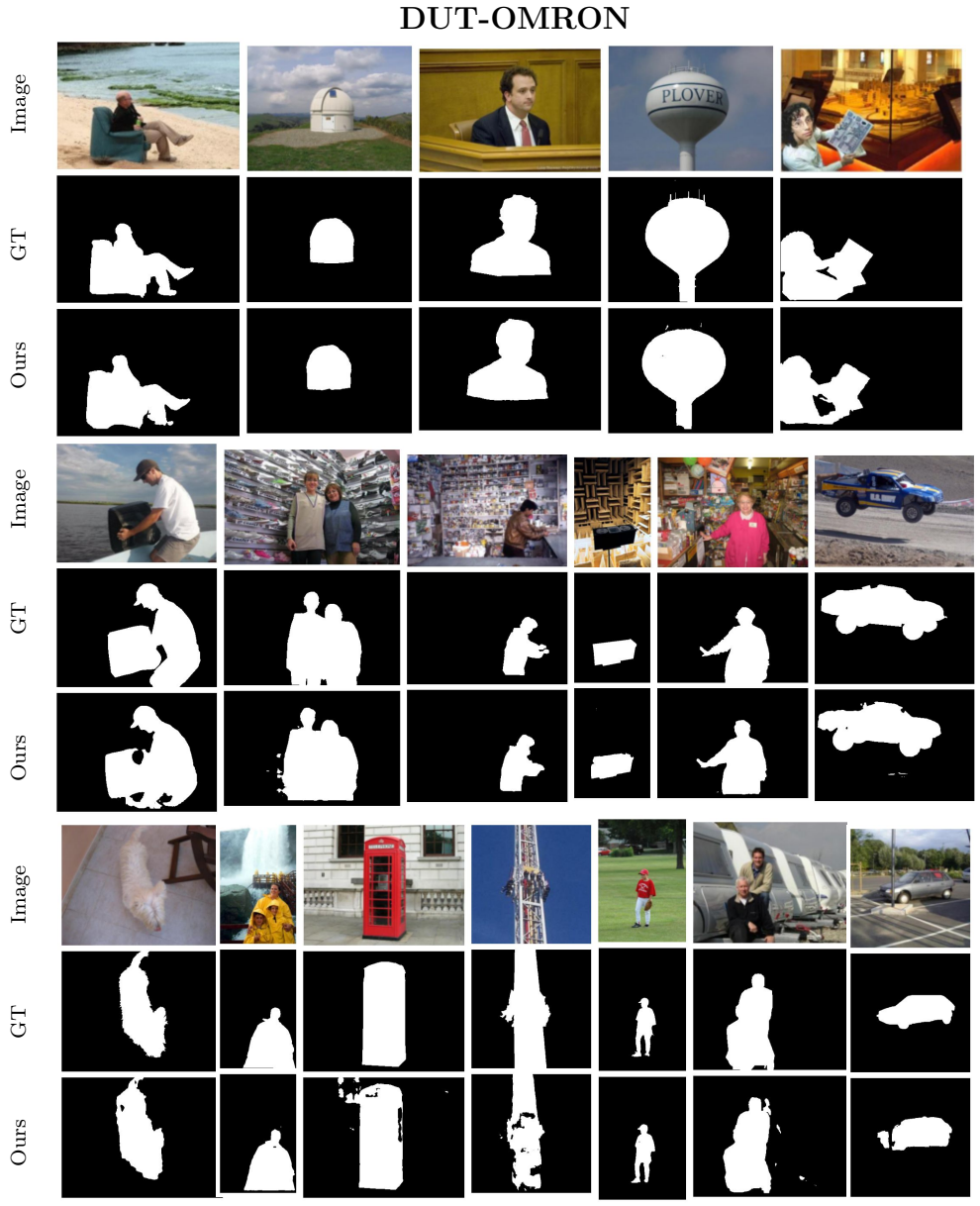


Figure C.2: **Qualitative Comparison on DUT-OMRON**. We train our model on a combined dataset of DAVIS, FBMS and STv2. Our model can segment both stationary and non-stationary objects and is robust enough to work on a wide range of classes

Figure C.3: **Qualitative Comparison on DUTS**. We train our model on a combined dataset of DAVIS, FBMS and STv2. We can segment a wide array of classes. Our model performs well on scenes where multiple objects are in the foreground

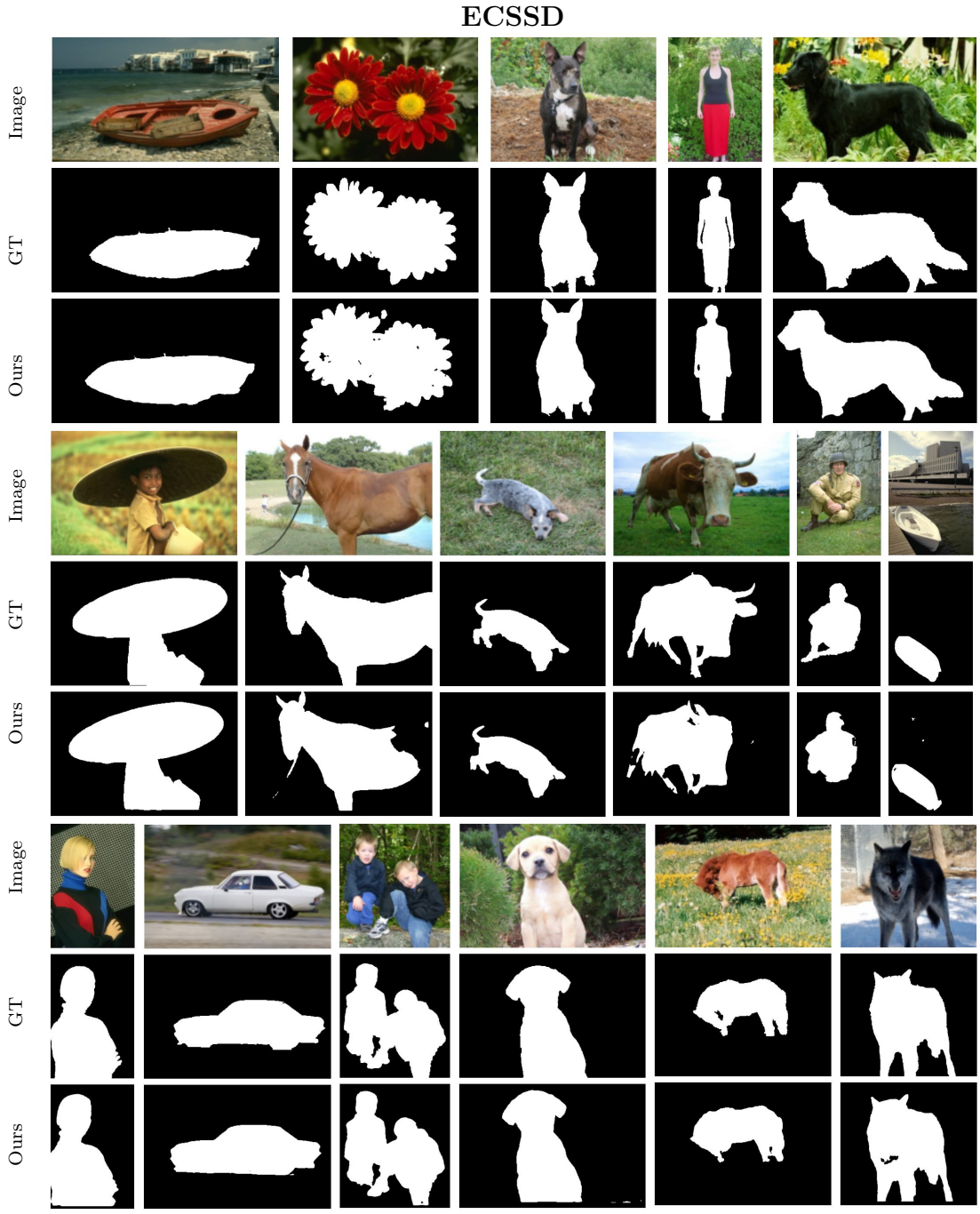


Figure C.4: **Qualitative Comparison on ECSSD**. We train our model on a combined dataset of DAVIS, FBMS and STv2. Our model can segment objects from different classes in complex poses

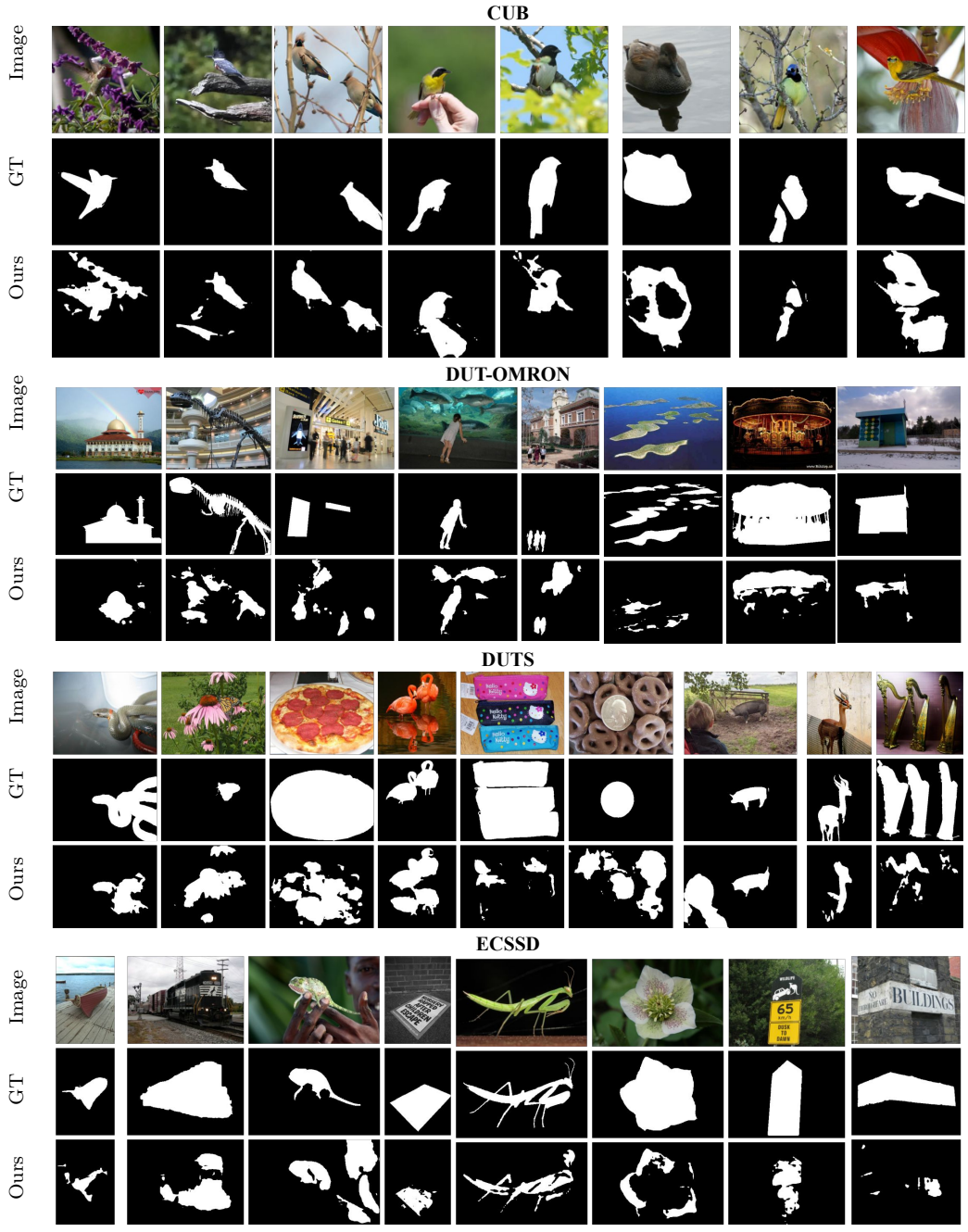


Figure C.5: **Qualitative Comparison of Failure Cases**. We train our model on a combined dataset of DAVIS, FBMS and STv2. Our method can extract salient object in various environments. The model has difficulty where the foreground object is ambiguous — when there are multiple prominent objects but only few are annotated as salient object. The model also has issues with predicting the object boundaries well for some instances

Appendix D

Unsupervised Multi-object Segmentation by Predicting Probable Motion Patterns Supplementary Material

In this supplementary material, we provide additional details for our loss function (appendix D.1), including detailed derivation steps, implementation details, and further discussion of the advantages. appendix D.3 specifies hyperparameters used and how they were selected. We conclude with additional ablation experiments (appendix D.4) and results (appendix D.5). Project page and code: https://www.robots.ox.ac.uk/~vgg/research/ppmp.

D.1 Loss derivation

The key part of our loss function is the likelihood of the optical flow $p(\mathbf{f} \mid \mathbf{m})$, which serves to evaluate how probable is a region of the optical flow carved out by the predicted masks for K regions. We assume that optical flow within a region depends only on the region itself and that other regions have no influence. Intuitively, this is a reasonable assumption, as in large, the movement/flow of an object does not depend on the background or other objects. Thus, enforcing this assumption encourages regions to correspond to objects.

The assessment of probability of optical flow within the region is based on the assumption that objects should be moving rigidly. We use an approximate parametric motion model (Eq. (2)). The parameters of the motion model θ abstract away unknown aspects such as scene geometry and camera intrinsics but enable to translate between assumed 3D rigid motion and 2D optical flow.

We assume that motion parameters $\theta_k \sim \mathcal{N}(\theta; \mu, \Sigma)$ come from a multivariate Gaussian prior. This choice enables expressing marginal-likelihood in closed-form.

We model the error of the approximate motion model as zero-mean isotropic Gaussian noise $\epsilon \sim \mathcal{N}(\epsilon; 0, \sigma^2 I)$.

Following the motion model (Eq. (2)), the optical flow $\mathbf{f}_k$ is an *affine combination* of Gaussian random variables. Using this observation, its distribution is

$$p(\mathbf{f}_k \mid \mathbf{m}_k) = \det(2\pi(P_k \Sigma P_k^\top + \sigma^2 I))^{-1/2} \cdot \exp(-\frac{1}{2}\mathbf{F}_k^\top (P_k \Sigma P_k^\top + \sigma^2 I)^{-1} \mathbf{F}_k), \quad \text{(D.9)}$$

where $\mathbf{F}_k = \mathbf{f}_k - \Pi_\mu(\Omega_k) + \Omega_k$ is the centered flow within the region k. This equation can be slightly simplified by considering its two troublesome parts, the determinant and the quadratic form inside the exponent. For the determinant, we note the following:

$$\begin{aligned}
\det(2\pi P_k \Sigma P_k^\top + 2\pi\sigma^2 I) &= (2\pi\sigma^2)^{2n_k} \det(1/\sigma^2 P_k \Sigma P_k^\top + I) \\
&= (2\pi\sigma^2)^{2n_k} \det(\Sigma) \det(1/\sigma^2 P_k^\top P_k + \Sigma^{-1}) \qquad (*) \\
&= \frac{(2\pi\sigma^2)^{2n_k}}{\det(\Lambda)} \det(\underbrace{1/\sigma^2 P_k^\top P_k + \Lambda}_{S_k}),
\end{aligned}$$

where in the line marked with (*) we apply matrix determinant lemma, and in the last line we substitute covariance $\Sigma^{-1} = \Lambda$ for the precision matrix. Similarly, the quadratic form in the exponent can be expanded

$$\begin{aligned}
\mathbf{F}_k^\top (P_k \Sigma P_k^\top + \sigma^2 I)^{-1} \mathbf{F}_k &= 1/\sigma^2 \mathbf{F}_k^\top (1/\sigma^2 P_k \Sigma P_k^\top + I)^{-1} \mathbf{F}_k \\
&= 1/\sigma^2 \mathbf{F}_k^\top \left(I - 1/\sigma^2 P_k (1/\sigma^2 P_k^\top P_k + \Lambda)^{-1} P_k^\top \right) \mathbf{F}_k \qquad (\dagger) \\
&= 1/\sigma^2 \mathbf{F}_k^\top \mathbf{F}_k - (1/\sigma^2)^2 \mathbf{F}_k^\top P_k (\underbrace{1/\sigma^2 P_k^\top P_k + \Lambda}_{S_k})^{-1} P_k^\top \mathbf{F}_k,
\end{aligned}$$

where in the line marked with (†) the Woodbury identity is applied. The optical

flow for the whole image is modeled as a joint of independent flow regions $\mathbf{f}_k$, giving the log-likelihood as

$$\begin{aligned}
&\log p(\mathbf{f} \mid \mathbf{m}) = \sum_k \log p(\mathbf{f}_k \mid \mathbf{m}_k) \\
&= -\frac{1}{2}\left(2\log(2\pi\sigma^2)\sum_k n_k + \sum_k \log\frac{\det S_k}{\det \Lambda} + {}^1\!/\!_{\sigma^2}\sum_k \mathbf{F}_k^\top\mathbf{F}_k - ({}^1\!/\!_{\sigma^2})^2\sum_k \mathbf{F}_k^\top P_k S_k^{-1} P_k^\top \mathbf{F}_k\right) \\
&= -\frac{1}{2}\left(2\log(2\pi\sigma^2)n + \sum_k \log\frac{\det S_k}{\det \Lambda} + {}^1\!/\!_{\sigma^2}\mathbf{F}^\top\mathbf{F} - ({}^1\!/\!_{\sigma^2})^2\sum_k \mathbf{F}^\top L_k P S_k^{-1} P^\top L_k \mathbf{F}\right),
\end{aligned} \tag{D.10}$$

where in the last line we introduced $n = \sum_k n_k = HW$, the number of pixels in the image. We also use product of selector matrices $L_k = R_k^\top R_k = \mathrm{diag}(\mathbf{m}_k, \mathbf{m}_k) = L_k^\top$ such that $\mathbf{F}_k^\top P_k = \mathbf{F}^\top L_k P$. This explicitly includes masks in the expression. Finally, we use the fact that regions partition the full image $\sum_k \mathbf{F}_k^\top\mathbf{F}_k = \sum_k\sum_i (\mathbf{F}_k)_i^2 = \mathbf{F}^\top\mathbf{F}$. We now manipulate eq. (D.10) using specific details of the motion model to arrive at expressions that are convenient to implement in code.

D.1.1 Implementation details

Translation-only likelihood We assume that translation along x and y directions is independent, such that θ prior is a zero-mean Gaussian with isotropic covariance $\tau^2 I$. $P_k^{\mathrm{tr}} = \mathrm{diag}(\mathbf{1}_{n_k}, \mathbf{1}_{n_k})$ and by extension $P^{\mathrm{tr}} = \mathrm{diag}(\mathbf{1}_n, \mathbf{1}_n)$. The matrix S_k simplifies to

$$S_k = {}^1\!/\!_{\sigma^2} P_k^\top P_k + \Lambda = {}^1\!/\!_{\sigma^2}\begin{pmatrix} \mathbf{1}_{n_k}^\top\mathbf{1}_{n_k} & 0 \\ 0 & \mathbf{1}_{n_k}^\top\mathbf{1}_{n_k}\end{pmatrix} + {}^1\!/\!_{\tau^2} I = {}^1\!/\!_{\sigma^2}(n_k + {}^{\sigma^2}\!/\!_{\tau^2})I\,,$$

such that

$$\det S_k = ({}^1\!/\!_{\sigma^2}(n_k + {}^{\sigma^2}\!/\!_{\tau^2}))^2, \quad \log\frac{\det S_k}{\det\Lambda} = 2\log\frac{n_k + {}^{\sigma^2}\!/\!_{\tau^2}}{{}^{\sigma^2}\!/\!_{\tau^2}}, \quad S_k^{-1} = ({}^1\!/\!_{\sigma^2}(n_k + {}^{\sigma^2}\!/\!_{\tau^2}))^{-1} I.$$

Writing $\mathbf{F}^\top = (\mathbf{u}^\top, \mathbf{v}^\top)$ to denote x and y components of the flow, respectively, the term reduces

$$
\begin{aligned}
(1/\sigma^2)^2 \sum_k \mathbf{F}^\top L_k P S_k^{-1} P^\top L_k \mathbf{F} &= 1/\sigma^2 \sum_k \mathbf{F}^\top L_k^\top P P^\top L_k \mathbf{F} \frac{1}{n_k + \sigma^2/\tau^2} \\
&= 1/\sigma^2 \sum_k \frac{1}{n_k + \sigma^2/\tau^2} \begin{bmatrix} \mathbf{u}^\top & \mathbf{v}^\top \end{bmatrix} \begin{bmatrix} \mathbf{m}_k \mathbf{m}_k^\top & \\ & \mathbf{m}_k \mathbf{m}_k^\top \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} \\
&= 1/\sigma^2 \sum_k \frac{1}{n_k + \sigma^2/\tau^2} \left((\mathbf{u}^\top \mathbf{m}_k)^2 + (\mathbf{v}^\top \mathbf{m}_k)^2 \right) \\
&= 1/\sigma^2 \sum_k \frac{n_k^2}{n_k + \sigma^2/\tau^2} (\bar{u}_k^2 + \bar{v}_k^2) \,,
\end{aligned}
$$

where in the last line we introduced mean flow $\bar{u}_k = n_k^{-1} \mathbf{u}^\top \mathbf{m}_k$ and $\bar{v}_k = n_k^{-1} \mathbf{v}^\top \mathbf{m}_k$. This gives the negative log-likelihood as

$$
\begin{aligned}
&\log p(\mathbf{f} \mid \mathbf{m}) = \sum_k \log p(\mathbf{f}_k \mid \mathbf{m}_k) \\
&= -\frac{1}{2} \left(2 \log(2\pi\sigma^2) n + \sum_k \log \frac{\det S_k}{\det \Lambda} + 1/\sigma^2 \mathbf{F}^\top \mathbf{F} - (1/\sigma^2)^2 \sum_k \mathbf{F}^\top L_k P S_k^{-1} P^\top L_k \mathbf{F} \right) \\
&= -n \log(2\pi\sigma^2) - \sum_k \log \frac{n_k + \sigma^2/\tau^2}{\sigma^2/\tau^2} - \frac{1}{2\sigma^2} \left(\mathbf{F}^\top \mathbf{F} - \sum_k \frac{n_k^2 (\bar{u}_k^2 + \bar{v}_k^2)}{n_k + \sigma^2/\tau^2} \right) \\
&= -n \log(2\pi\sigma^2) - \sum_k \log \frac{n_k + \sigma^2/\tau^2}{\sigma^2/\tau^2} - \frac{1}{2\sigma^2} \sum_{i=1}^n \left(u_i^2 + v_i^2 - \sum_k \frac{n_k (\bar{u}_k^2 + \bar{v}_k^2)}{n_k + \sigma^2/\tau^2} (\mathbf{m}_k)_i \right) .
\end{aligned}
\tag{D.11}
$$

In our implementation, we extend this equation further. Writing $w_k = 1 - \sqrt{\frac{\sigma^2/\tau^2}{n_k + \sigma^2/\tau^2}}$, we note that the following sum is equivalent

$$
\begin{aligned}
\sum_{i=1}^{n}(u_i - \sum_k \bar{u}_k w_k (\mathbf{m}_k)_i)^2 &= \sum_{i=1}^{n}(u_i^2 - \sum_k 2u_i \bar{u}_k w_k (\mathbf{m}_k)_i + \sum_k (\mathbf{m}_k)_i w_k^2 \bar{u}_k^2) \\
&= \sum_{i=1}^{n} u_i^2 - \sum_k 2\bar{u}_k w_k \sum_{i=1}^{n} u_i (\mathbf{m}_k)_i + \sum_k w_k^2 \bar{u}_k^2 \sum_{i=1}^{n} (\mathbf{m}_k)_i \\
&= \sum_{i=1}^{n} u_i^2 - \sum_k 2\bar{u}_k^2 w_k n_k + \sum_k w_k^2 \bar{u}_k^2 n_k \\
&= \sum_{i=1}^{n} u_i^2 - \sum_k \bar{u}_k^2 n_k (2w_k - w_k^2) \\
&= \sum_{i=1}^{n} u_i^2 - \sum_k \bar{u}_k^2 \frac{n_k}{n_k + \sigma^2/\tau^2} \sum_{i=1}^{n} (\mathbf{m}_k)_i \\
&= \sum_{i=1}^{n} \left(u_i^2 - \sum_k \frac{n_k \bar{u}_k^2 (\mathbf{m}_k)_i}{n_k + \sigma^2/\tau^2} \right),
\end{aligned}
$$

where in the first line we make use of the fact that masks are one-hot $(\mathbf{m})_i \in \{0,1\}^k$, thus only a single term in the sums over k is non-zero, i.e. $(\sum_k w_k \bar{u}_k (\mathbf{m}_k)_i)^2 = \sum_k w_k^2 \bar{u}_k^2 (\mathbf{m}_k)_i)$. Using the above insight, the log-likelihood is

$$
\begin{aligned}
\log p(\mathbf{f} \mid \mathbf{m}) = &-n \log(2\pi\sigma^2) - \sum_k \log \frac{n_k + \sigma^2/\tau^2}{\sigma^2/\tau^2} \\
&- \frac{1}{2\sigma^2} \sum_{i=1}^{n} \left((u_i - \sum_k \bar{u}_k w_k (\mathbf{m}_k)_i)^2 + (v_i - \sum_k \bar{v}_k w_k (\mathbf{m}_k)_i)^2 \right), \quad \text{(D.12)}
\end{aligned}
$$

where

$$
n_k = \sum_{i=1}^{n} (\mathbf{m}_k)_i \,, \quad w_k = 1 - \sqrt{\frac{\sigma^2/\tau^2}{n_k + \sigma^2/\tau^2}} \,.
$$

We then replace $\mathbf{m}$ with $\hat{\mathbf{m}}$ from the Gumbel-Softmax approximation.

Affine motion likelihood. For the affine motion model, the full covariance matrix Σ prevents significant further simplification. Instead, we transform the log-likelihood (eq. (D.10)) to an equivalent form involving only 3×3 matrices, for which the required determinant and inverse can be calculated analytically. To that end, we introduce the following auxiliary variables:

$$
G_k = \begin{bmatrix} \mathbf{x}_k & \mathbf{y}_k & \mathbf{1}_{n_k} \end{bmatrix}, \quad P_k = \begin{bmatrix} G_k & 0 \\ 0 & G_k \end{bmatrix}, \quad \Sigma^{-1} = \Lambda = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}.
$$

Then S_k is

$$S_k = {}^1\!/\!_{\sigma^2} P_k^\top P_k + \Lambda = \begin{bmatrix} {}^1\!/\!_{\sigma^2} G_k^\top G_k + \alpha & \beta \\ \gamma & {}^1\!/\!_{\sigma^2} G_k^\top G_k + \delta \end{bmatrix},$$

with

$$G_k^\top G_k = \begin{pmatrix} \mathbf{x}_k^\top \mathbf{x}_k & \mathbf{x}_k^\top \mathbf{y}_k & \mathbf{x}_k^\top \mathbf{1}_{n_k} \\ \mathbf{x}_k^\top \mathbf{y}_k & \mathbf{y}_k^\top \mathbf{y}_k & \mathbf{y}_k^\top \mathbf{1}_{n_k} \\ \mathbf{x}_k^\top \mathbf{1}_{n_k} & \mathbf{y}_k^\top \mathbf{1}_{n_k} & \mathbf{1}_{n_k}^\top \mathbf{1}_{n_k} \end{pmatrix}.$$

Using this, the determinant is

$$\det S_k = \det({}^1\!/\!_{\sigma^2} G_k^\top G_k + \alpha - \beta({}^1\!/\!_{\sigma^2} G_k^\top G_k + \delta)^{-1}\gamma) \det({}^1\!/\!_{\sigma^2} G_k^\top G_k + \delta).$$

Similarly, the inverse is then

$$\begin{aligned} S_k^{-1} &= \begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix}, \quad \text{where} \\ D_k &= ({}^1\!/\!_{\sigma^2} G_k^\top G_k + \delta - \gamma({}^1\!/\!_{\sigma^2} G_k^\top G_k + \alpha)^{-1}\beta) \\ C_k &= -D_k\gamma({}^1\!/\!_{\sigma^2} G_k^\top G_k + \alpha)^{-1} \\ B_k &= -({}^1\!/\!_{\sigma^2} G_k^\top G_k + \alpha)^{-1}\beta D_k \\ A_k &= ({}^1\!/\!_{\sigma^2} G_k^\top G_k + \alpha)^{-1} - B_k\gamma({}^1\!/\!_{\sigma^2} G_k^\top G_k + \alpha)^{-1}, \text{ such that} \\ \mathbf{h}_k &= \begin{pmatrix} \mathbf{u}_k^\top \mathbf{x}_k & \mathbf{u}_k^\top \mathbf{y}_k & \mathbf{u}_k^\top \mathbf{1}_{n_k} \end{pmatrix} \\ \mathbf{r}_k &= \begin{pmatrix} \mathbf{v}_k^\top \mathbf{x}_k & \mathbf{v}_k^\top \mathbf{y}_k & \mathbf{v}_k^\top \mathbf{1}_{n_k} \end{pmatrix} \\ \mathbf{F}_k^\top P_k S_k^{-1} P_k^\top \mathbf{F}_k &= \mathbf{h}_k A_k \mathbf{r}_k^\top + \mathbf{r}_k C_k \mathbf{h}_k^\top + \mathbf{h}_k B_k \mathbf{r}_k^\top + \mathbf{r}_k D_k \mathbf{r}_k^\top. \end{aligned}$$

We implement inner products under Gumbel-Softmax as $\mathbf{a}_k^\top \mathbf{b}_k = \sum_{i=1}^n (\mathbf{a})_i (\mathbf{b})_i (\hat{\mathbf{m}}_k)_i$, for some vectors $\mathbf{a}, \mathbf{b}$. The coordinate vectors $\mathbf{x}_k, \mathbf{y}_k$ have the origin set to the centroid of the predicted region $\begin{pmatrix} x_{c,k} \\ y_{c,k} \end{pmatrix} = n_k^{-1} \begin{pmatrix} \mathbf{x}^\top \mathbf{m}_k \\ \mathbf{y}^\top \mathbf{m}_k \end{pmatrix}$. The expressions can then be substituted back to eq. (D.10). We show implementation in algorithm 1.

D.1.2 Further justification

We consider whether the inclusion of the prior on the motion parameters offers any benefits. Consider a simple translation-only model. Since objects are only translating, each pixel in a region should be very close to the mean translation

Algorithm 1 Implementation of negative flow likelihood $-\log p(\mathbf{f} \mid \mathbf{m})$ under affine motion prior. Key quantities in the inner loop are underlined.

procedure NLL(mean μ, covariance Σ, variance σ^2, flow $\mathbf{f}$, masks $\mathbf{m}_k$, height H, width W)
  $(\mu_1\ \mu_2\ \mu_3\ \mu_4\ \mu_5\ \mu_6) \leftarrow \mu$
  $\mathbf{x}, \mathbf{y} \leftarrow \text{lattice}(H, W)$
  $\left[\begin{smallmatrix}\mathbf{u}\\ \mathbf{v}\end{smallmatrix}\right] \leftarrow \mathbf{f}$
  $\Lambda \leftarrow \Sigma^{-1}$
  $\left[\begin{smallmatrix}\alpha & \beta\\ \gamma & \delta\end{smallmatrix}\right] \leftarrow \Lambda$
  for all k regions **do**
    $n_k \leftarrow \sum_i (\mathbf{m}_k)_i$
    $\hat{\mathbf{x}}_k \leftarrow \mathbf{x} - n_k^{-1} \mathbf{x}^\top \mathbf{m}_k$ ▷ Set origin to centroid
    $\hat{\mathbf{y}}_k \leftarrow \mathbf{y} - n_k^{-1} \mathbf{y}^\top \mathbf{m}_k$
    $\mathbf{x}_k^\top \mathbf{x}_k \leftarrow \sum_i (\hat{\mathbf{x}}_k)_i^2 (\mathbf{m}_k)_i$
    $\mathbf{x}_k^\top \mathbf{y}_k \leftarrow \sum_i (\hat{\mathbf{x}}_k)_i (\hat{\mathbf{y}}_k)_i (\mathbf{m}_k)_i$
    $\mathbf{y}_k^\top \mathbf{y}_k \leftarrow \sum_i (\hat{\mathbf{y}}_k)_i^2 (\mathbf{m}_k)_i$
    $\mathbf{x}_k^\top \mathbf{1}_{n_k} \leftarrow \sum_i (\hat{\mathbf{x}}_k)_i (\mathbf{m}_k)_i$
    $\mathbf{y}_k^\top \mathbf{1}_{n_k} \leftarrow \sum_i (\hat{\mathbf{y}}_k)_i (\mathbf{m}_k)_i$
    $G_k^\top G_k \leftarrow \begin{pmatrix} \mathbf{x}_k^\top \mathbf{x}_k & \mathbf{x}_k^\top \mathbf{y}_k & \mathbf{x}_k^\top \mathbf{1}_{n_k} \\ \mathbf{x}_k^\top \mathbf{y}_k & \mathbf{y}_k^\top \mathbf{y}_k & \mathbf{y}_k^\top \mathbf{1}_{n_k} \\ \mathbf{x}_k^\top \mathbf{1}_{n_k} & \mathbf{y}_k^\top \mathbf{1}_{n_k} & n_k \end{pmatrix}$
    $D_k \leftarrow (1/\sigma^2 G_k^\top G_k + \delta - \gamma(1/\sigma^2 G_k^\top G_k + \alpha)^{-1}\beta)$
    $C_k \leftarrow -D_k \gamma (1/\sigma^2 G_k^\top G_k + \alpha)^{-1}$
    $B_k \leftarrow -(1/\sigma^2 G_k^\top G_k + \alpha)^{-1} \beta D_k$
    $A_k \leftarrow (1/\sigma^2 G_k^\top G_k + \alpha)^{-1} - B_k \gamma (1/\sigma^2 G_k^\top G_k + \alpha)^{-1}$
    $\mathbf{u}_k^\top \mathbf{x}_k \leftarrow \sum_i ((\mathbf{u})_i)((\mathbf{x})_i - x_{c,k})(\mathbf{m}_k)_i$
    $\hat{\mathbf{u}}_k \leftarrow \mathbf{u} - ((\mu_1 - 1)\hat{\mathbf{x}} + \mu_2 \hat{\mathbf{y}} + \mu_3)$ ▷ Center flow according to mean motion
    $\hat{\mathbf{v}}_k \leftarrow \mathbf{v} - ((\mu_5 - 1)\hat{\mathbf{y}} + \mu_4 \hat{\mathbf{x}} + \mu_6)$
    $\underline{\mathbf{F}_k^\top \mathbf{F}_k} \leftarrow \sum_i (\hat{\mathbf{u}}_k)_i^2 (\mathbf{m}_k)_i + \sum_i (\hat{\mathbf{v}}_k)_i^2 (\mathbf{m}_k)_i$
    $\mathbf{u}_k^\top \mathbf{x}_k \leftarrow \sum_i (\hat{\mathbf{u}}_k)_i (\hat{\mathbf{x}}_k)_i (\mathbf{m}_k)_i$
    $\mathbf{u}_k^\top \mathbf{y}_k \leftarrow \sum_i (\hat{\mathbf{u}}_k)_i (\hat{\mathbf{y}}_k)_i (\mathbf{m}_k)_i$
    $\mathbf{u}_k^\top \mathbf{1}_{n_k} \leftarrow \sum_i (\hat{\mathbf{u}}_k)_i (\mathbf{m}_k)_i$
    $\mathbf{h}_k \leftarrow \begin{pmatrix} \mathbf{u}_k^\top \mathbf{x}_k & \mathbf{u}_k^\top \mathbf{y}_k & \mathbf{u}_k^\top \mathbf{1}_{n_k} \end{pmatrix}$
    $\mathbf{v}_k^\top \mathbf{x}_k \leftarrow \sum_i (\hat{\mathbf{v}}_k)_i (\hat{\mathbf{x}}_k)_i (\mathbf{m}_k)_i$
    $\mathbf{v}_k^\top \mathbf{y}_k \leftarrow \sum_i (\hat{\mathbf{v}}_k)_i (\hat{\mathbf{y}}_k)_i (\mathbf{m}_k)_i$
    $\mathbf{v}_k^\top \mathbf{1}_{n_k} \leftarrow \sum_i (\hat{\mathbf{v}}_k)_i (\mathbf{m}_k)_i$
    $\mathbf{r}_k \leftarrow \begin{pmatrix} \mathbf{v}_k^\top \mathbf{x}_k & \mathbf{v}_k^\top \mathbf{y}_k & \mathbf{v}_k^\top \mathbf{1}_{n_k} \end{pmatrix}$
    $\underline{\mathbf{F}_k^\top P_k S_k^{-1} P_k^\top \mathbf{F}_k} \leftarrow \mathbf{h}_k A_k \mathbf{r}_k^\top + \mathbf{r}_k C_k \mathbf{h}_k^\top + \mathbf{h}_k B_k \mathbf{r}_k^\top + \mathbf{r}_k D_k \mathbf{r}_k^\top$
    $\underline{\det S_k} \leftarrow \det(1/\sigma^2 G_k^\top G_k + \alpha - \beta(1/\sigma^2 G_k^\top G_k + \delta)^{-1}\gamma) \det(1/\sigma^2 G_k^\top G_k + \delta)$
  return $HW \log(2\pi\sigma^2) + \frac{1}{2}\sum_k \log \frac{\det S_k}{\det \Lambda} + \frac{1}{2\sigma^2}\sum_k (\mathbf{F}_k^\top \mathbf{F}_k - \mathbf{F}_k^\top P_k S_k^{-1} P_k^\top \mathbf{F}_k)$

of that region. We can assess the mean for a region as $\left[\begin{smallmatrix}\bar{u}_k\mathbf{m}_k\\ \bar{v}_k\mathbf{m}_k\end{smallmatrix}\right]$, considering some variance σ^2 around it:

$$\begin{aligned}\log\hat{p}(\mathbf{f}\mid\mathbf{m}) &= \log\mathcal{N}([\begin{smallmatrix}\mathbf{u}\\ \mathbf{v}\end{smallmatrix}];\sum_k\left[\begin{smallmatrix}\bar{u}_k\mathbf{m}_k\\ \bar{v}_k\mathbf{m}_k\end{smallmatrix}\right],\sigma^2 I)\\ &= -n\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left((u_i-\sum_k\bar{u}_k(\mathbf{m}_k)_i)^2+(v_i-\sum_k\bar{v}_k(\mathbf{m}_k)_i)^2\right).\end{aligned}$$

Such model, up to a scaling factor, was already considered for features [Choudhury et al. 2021] and optical flow [Choudhury et al. 2022]. This expression for $\log\hat{p}(\mathbf{f}\mid\mathbf{m})$ can be compared with our version of translation only model eq. (D.12). By considering the prior on the motion parameters, we introduce a weighing factor w_k for each mean $\bar{u}_k\mathbf{m}_k$, which discounts the contribution of smaller segments. Similarly, the term $\sum_k\log\frac{n_k+\sigma^2/\tau^2}{\sigma^2/\tau^2}\approx\sum_k\log n_k$ encourages larger masks, since $\sum_k n_k = n$. The prior helps to encode that larger regions should be preferred.

D.2 MovingClevrTex and MovingClevr

We extend the implementation of [Karazija et al. 2021] to generate video datasets of CLEVR and ClevrTex scenes. We follow original sampling set up of ClevrTex – each scene contains a random arrangement of 3–10 objects. We uniformly choose between scenarios where a single random objects is given initial motion, two random objects are provided initial motion or all objects are moving. We sample a random initial translation in XY plane for the object between keyframes 0 and 3, which builds momentum. Physics simulation takes over from keyframe 4. Mass of each object is set to equal its scale (numerically), and we use value of 0.1 for the 'bounciness' parameters. This results in objects sliding, rotating due to shape and friction, and colliding. Collisions can make other objects move. Each simulation is 5 frames long and we render keyframes 4 to 8.

We sample 10000 scenes for MovingClevrTex, which gives the same number of frames as the original ClevrTex (where each scene had only single frame). For MovingClevr, we sample 5000 scenes. 1000 and 500 scenes are kept as validation for MovingClevrTex and for MovingClevr, respectively. We use the same rendering and lighting parameters as in [Karazija et al. 2021], except we slightly reduce the scale of the surface displacement mapping for the background.

This reduces the visibility of clipping of the detailed object geometry with the background geometry, which might occur due to physics simulation working on simplified meshes.

D.3 Hyperparameters

Our method can use any segmentation network Φ. We employ the recent Mask2Former[1] architecture, using 6-layer CNN backbone from [Locatello et al. 2020; Kipf et al. 2022] for simulated datasets and ResNet-18 for KITTI in the main experiments. We also experiment with Swin-tiny transformer as the backbone, as it offers balanced performance in both visually simple and complex settings. We ablate these choices in appendix D.4.

The networks are trained with AdamW [Loshchilov and Hutter 2018], with a learning rate of 3×10^{-6} and batch size of 32, for 250k iterations. We employ gradient clipping when the 2-norm exceeds 0.01 and linear learning rate warm-up for the first 5k iterations to stabilize the training. The learning rate is reduced by a factor of 10 after 200k iterations. When training with warping loss on MOVi datasets, we found it beneficial to train for longer, for 500k iterations, reducing the learning rate by factor of 10 also after 400k iterations.

We found it beneficial to linearly anneal β from 0.1 to -0.1 over 5k iterations, encouraging the network to explore initially but focus on low-entropy distributions in the end. We found this had the effect of encouraging the network to assign background pixels to a single slot.

For the prior, we set σ^2 (eq. (D.10)) to 0.5 and use $\mu = (1\ 0\ 0\ 0\ 1\ 0)^\top$. We use the following covariance matrix

$$\Sigma = \begin{pmatrix} 0.005 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.05 & 0 & 0 & 0 & 0 \\ 0 & 0 & 15 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.05 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.005 & 0 \\ 0 & 0 & 0 & 0 & 0 & 15 \end{pmatrix}.$$

[1] Code available from https://github.com/facebookresearch/Mask2Former.

D.3.1 Settings in ablations

When experimenting with translation-only model, we use the same parameters and settings where possible. We set τ^2, so $0.5, 16.0, 34.0$ for CLEVR/CLEVRTEX, MOVi-A, and MOVi-C, respectively (values picked from specialized covariance matrices, see appendix D.4).

For GNM [Jiang and Ahn 2020], we use implementation and parameters described in [Karazija et al. 2021]. Owning to our loss being lower bound on log-probability, we simply add our loss to existing ELBO loss with hyperparameters above, only changing $\sigma^2 = 0.1$. We hypothesize that lower noise model is beneficial, as it provides stronger learning signal in the early stages when reconstruction is noisy due to untrained VAEs. The overfitting to errors of the motion approximation is handled, instead, by the networks balancing between appearance reconstruction and motion explanation objectives.

For SA [Locatello et al. 2020], we also use implementation of [Karazija et al. 2021]. SA loss is multiplied by 100 before adding our formulation.

D.3.2 Settings in KITTI

For experiments on KITTI, we replace the backbone to ResNet-18 to match prior work. We reduce batch size to 8, increase learning rate to 10^{-4}. Learning rate is linearly warmed up for 10 iterations, and reduced by a factor of 10 after 5500 iterations. We employ backbone learning rate multiplier of 0.1. All other settings as before.

D.3.3 Model parameters of comparisons

Here we describe implementation and hyperparameters used for comparative methods in our experiments.

SAVi [Kipf et al. 2022]. We follow the parameters given for *conditional* SAVi-S in their code repository[2], except to make SAVi unconditional we unset the conditioning key and make the slots to be learnable parameters.

[2]Code available from https://github.com/google-research/slot-attention-video.

SCALOR [Jiang et al. 2020]. We use the optimized SCALOR parameters mentioned by SAVi [Kipf et al. 2022] to train SCALOR. Particularly we use the MOVI dataset parameters to train MOVi-A and MOVI++ parameters to train MOVi-C.

GWM [Choudhury et al. 2022]. For GWM, we follow the parameters mentioned in the paper, except we do not employ spectral clustering and match the number of components to our settings for each dataset.

D.4 Additional ablations

Segmentation network. We study the effect of segmentation network in table D.1a. We find that using a much simpler U-Net [Ronneberger et al. 2015] architecture is beneficial on MOVi-A which contains visually plain scenes. The U-Net architecture, however, results in performance degradation on visually complex data.

We also consider a version of Mask2Former architecture that uses much deeper backbone, using Resnet-50 and Resnet-18, instead. We find that the deeper backbones leads to similar performance, indicating that our formulation is not architecture-specific. Finally, we change to MaskFormer architecture, matching the network architecture used in [Choudhury et al. 2022], and use smaller Swin-tiny backbone. We find that our loss formulation leads to improved performance still.

Covariance matrix. We investigate whether further improvements are possible using specialised versions of the prior. To that end, we offset the mean translation prior to account for the dominantly downward motion of objects on MOVi-A/MOVi-C. We use $\mu_{\text{MOVi-A}} = (1\ 0\ 0\ 0\ 1\ 1.5)^\top$ and $\mu_{\text{MOVi-C}} = (1\ 0\ 0\ 0\ 1\ 1.8)^\top$, respectively.[3] We use the following specialized covariance matrices:

[3] In our experiments, Y axis is pointing down and X is pointing right.

Table D.1: Supplementary ablations for our methods. We show the impact segmentation networks have for different datasets table D.1a. We further study the impact of our tuned covariance matrix in table D.1b. Results with post-processing applied.

(a) Choice of Model Architecture

		MOVi-A		MOVi-C		MovingClevrTex	
Architecture	**# Params**	**FG-ARI↑**	**mIoU↑**	**FG-ARI↑**	**mIoU↑**	**FG-ARI↑**	**mIoU↑**
M2F (Swin-tiny)	47M	83.48	72.61	58.59	35.67	88.80	69.62
M2F (ResNet50)	44M	83.44	68.06	60.32	34.80	90.40	67.07
M2F (ResNet18)	31M	84.04	67.48	60.84	35.69	90.31	67.33
MF (Swin-tiny)	44M	81.78	71.28	54.45	33.67	71.07	51.06
U-Net	31M	90.79	82.85	60.28	26.62	87.03	39.66

(b) Choice of Covariance Marix

	MOVi-A		MOVi-C	
Σ	**FG-ARI↑**	**mIoU↑**	**FG-ARI↑**	**mIoU↑**
Generic	82.32	71.70	58.12	35.79
Tuned	83.48	72.61	58.59	35.67

$$\Sigma_{\text{MOVi-A}} = \begin{pmatrix} 0.006 & -0.00004 & 0 & 0.00004 & 0.001 & 0 \\ -0.00004 & 0.04 & 0 & -0.01 & -0.00008 & 0 \\ 0 & 0 & 16 & 0 & 0 & 0 \\ 0.00004 & -0.01 & 0 & 0.04 & 0.00004 & 0 \\ 0.001 & -0.00008 & 0 & 0.00004 & 0.006 & 0 \\ 0 & 0 & 0 & 0 & 0 & 14 \end{pmatrix}$$

$$\Sigma_{\text{MOVi-C}} = \begin{pmatrix} 0.02 & 0.00002 & 0 & 0.000009 & 0.002 & 0 \\ 0.00002 & 0.03 & 0 & -0.009 & -0.000006 & 0 \\ 0 & 0 & 36 & 0 & 0 & 0 \\ 0.000009 & -0.009 & 0 & 0.04 & -0.00007 & 0 \\ 0.002 & -0.000006 & 0 & -0.00007 & 0.02 & 0 \\ 0 & 0 & 0 & 0 & 0 & 34 \end{pmatrix}$$

We obtain the dataset specific covariance matrices by forming initial estimates using a method described below using only optical flow. We then overwrite the entries to encode our belief that translation should be independent. Finally, we further tuned the values through by taking one search step for MOVi-A and MOVi-C each. In our experiments, we found that increasing diagonal elements (variances) and

decreasing off-diagonal elements produced slightly better results.

Initial covariance estimation. We form initial estimate for the dataset-specific covariance matrices used in ablation only to start hyperparameter search in a sensible range. This method relies on observation that to form an estimate (1) all objects from a frame are not required – only some are sufficient. Furthermore, for the selected object candidates, (2) precise boundaries are not necessary. The method is as follows:

1. We extract discontinuities from the flow using Sobel filtering and treat these as flow edges.
2. We then only consider regions where the optical flow is larger than zero, identifying foreground.
3. We subtract edge pixels from the candidate foreground mask. This attempts to disconnect any overlapping objects using discontinuity in optical flow.
4. We run connected components algorithm to identify candidate object regions.
5. Within each region, using the motion model (Eq. (2)) we estimate $\hat{\theta}$ by forming least-squares solution. We only considered estimates from regions larger than 100 pixels (for numerical stability) and where the residual error was within the 90% percentile.
6. Initial covariance estimate $\hat{\Sigma}$ is formed by calculating sample covariance over the combined set of inliers and extra n no-motion values to account for stationary regions.

We apply this method on a subset of the data. n is the size of the subset.

We find using the specialized settings above give slight improvement on most metrics (table D.1b), indicating that using more appropriate prior for the data further improves results.

D.5 Additional results

Expanded results on Clevr and ClevrTex. In table D.2 we show expanded version of the results on CLEVR and CLEVRTEX benchmarks.

Table D.2: Expanded benchmark results on CLEVR, CLEVRTEX, CAMO, and OOD comparing FG-ARI and mIoU metrics. Results are a mean of 3 seed ($\pm\sigma$). Methods above the line are trained on single images, while methods below train on videos.† – indicates post-processing.

	CLEVR		CLEVRTEX		OOD		CAMO	
Model	**FG-ARI↑**	**mIoU↑**	**FG-ARI↑**	**mIoU↑**	**FG-ARI↑**	**mIoU↑**	**FG-ARI↑**	**mIoU↑**
SPAIR [Crawford and Pineau 2019]	77.13± 1.92	65.95± 4.02	0.00± 0.00	0.00±0.00	0.00± 0.00	0.00±0.00	0.00± 0.00	0.00±0.00
SPAIR†	77.05± 1.96	66.87± 9.65	0.00± 0.00	0.00±0.00	0.00± 0.00	0.00±0.00	0.00± 0.00	0.00±0.00
SPACE [Z. Lin et al. 2020b]	22.75±14.04	26.31±12.93	17.53± 4.13	9.14±3.46	12.71± 3.44	6.87±3.32	10.55± 2.09	8.67±3.50
SPACE†	22.74±14.03	27.00±13.69	17.52± 4.12	9.68±4.10	12.71± 3.44	7.20±3.75	10.54± 2.08	9.25±3.95
GenV2 [Engelcke et al. 2021]	57.90±20.38	9.48± 0.55	31.19±12.41	7.93±1.53	29.04±11.23	8.74±1.64	29.60±12.84	7.49±1.67
GenV2†	57.78±21.12	10.76± 1.39	30.55±14.27	9.04±0.63	28.41±13.20	9.96±0.70	29.19±14.55	8.40±1.00
MN [Smirnov et al. 2021]	72.12± 0.64	56.81±0.40	38.31± 0.70	10.46±0.10	37.29± 1.04	12.13±0.19	31.52± 0.87	8.79±0.15
MN†	72.08± 0.62	57.61±0.40	38.34± 0.73	10.34±0.12	37.28± 1.07	11.97±0.21	31.54± 0.87	8.77±0.18
MONet [C. P. Burgess et al. 2019]	54.47±11.41	30.66±14.87	36.66± 0.87	19.78±1.02	32.97± 1.00	19.30±0.37	12.44± 0.73	10.52±0.38
MONet†	61.36± 7.33	45.61± 4.80	35.64± 1.17	23.59±0.29	31.51± 1.46	23.04±0.52	9.94± 0.50	11.31±0.30
SA [Locatello et al. 2020]	95.89± 2.37	36.61±24.83	62.40± 2.23	22.58±2.07	58.45± 1.87	20.98±1.59	57.54± 1.01	19.83±1.41
SA†	94.88± 1.67	37.68±26.56	61.60± 2.29	21.96±1.79	57.41± 1.92	20.60±1.45	56.85± 1.12	19.42±1.42
IODINE [Greff et al. 2019]	93.81± 0.76	45.14±17.85	59.52± 2.20	29.17±0.75	53.20± 2.55	26.28±0.85	36.31± 2.57	17.52±0.75
IODINE†	93.68± 0.83	44.20±18.67	60.63± 2.50	29.40±1.10	54.92± 2.24	27.96±0.81	38.29± 1.40	18.87±0.52
eMORL [Emami et al. 2021]	93.25± 3.24	50.19±22.56	55.62± 2.12	30.17±2.60	49.21± 2.69	25.03±1.99	37.66± 8.41	19.13±4.88
eMORL†	93.09± 2.68	49.28±24.28	58.59± 1.96	31.64±2.22	51.97± 2.44	26.91±1.69	43.83± 7.34	22.40±4.35
DTI-S [Monnier et al. 2021]	89.54± 1.44	48.74± 2.17	79.90± 1.37	33.79±1.30	73.67± 0.98	32.55±1.08	72.90± 1.89	27.54±1.55
DTI-S†	89.86± 1.78	53.38± 3.51	79.86± 1.36	32.20±1.49	73.60± 0.97	30.74±1.22	72.89± 1.88	26.30±1.57
GNM [Jiang and Ahn 2020]	65.05± 4.19	59.92± 3.72	53.37± 0.67	42.25±0.18	48.43± 0.86	40.84±0.30	15.73± 0.89	17.56±0.74
GNM†	65.67± 4.23	63.38± 3.76	53.38± 0.67	44.30±0.19	48.44± 0.86	42.87±0.28	15.72± 0.89	18.53±0.75
SAVi [Kipf et al. 2022]	—	—	49.54	31.88	42.68	30.31	42.67	29.60
Ours	91.69± 0.30	66.70± 0.32	90.80± 0.22	55.07±0.44	76.01± 0.56	46.84±0.20	72.78± 1.31	42.30±1.09
Ours †	**95.94**± 0.43	**84.86**± 4.06	**92.61**± 0.22	**77.67**±0.25	**78.24**± 0.43	**55.54**±0.44	**77.43**± 0.86	**56.43**±0.80

Qualitative results on MOVi-A and MOVi-C. In fig. D.1 and D.2 we show additional qualitative results on MOVi-A and MOVi-C respectively. Following the results in main paper, the segments discovered by our method are semantically meaningful. Our object boundaries are of higher quality than the comparable methods. GWM suffers from oversegmentation of the objects. SCALOR has dificulty with complex datasets, such as MOVi-C as observed in fig. D.2. SAVI's object boundaries do not conform to object shape. We also provide additional failure cases of our model in fig. D.3. Our model has difficulty with objects that have complex motion for our affine formulation to model ably.

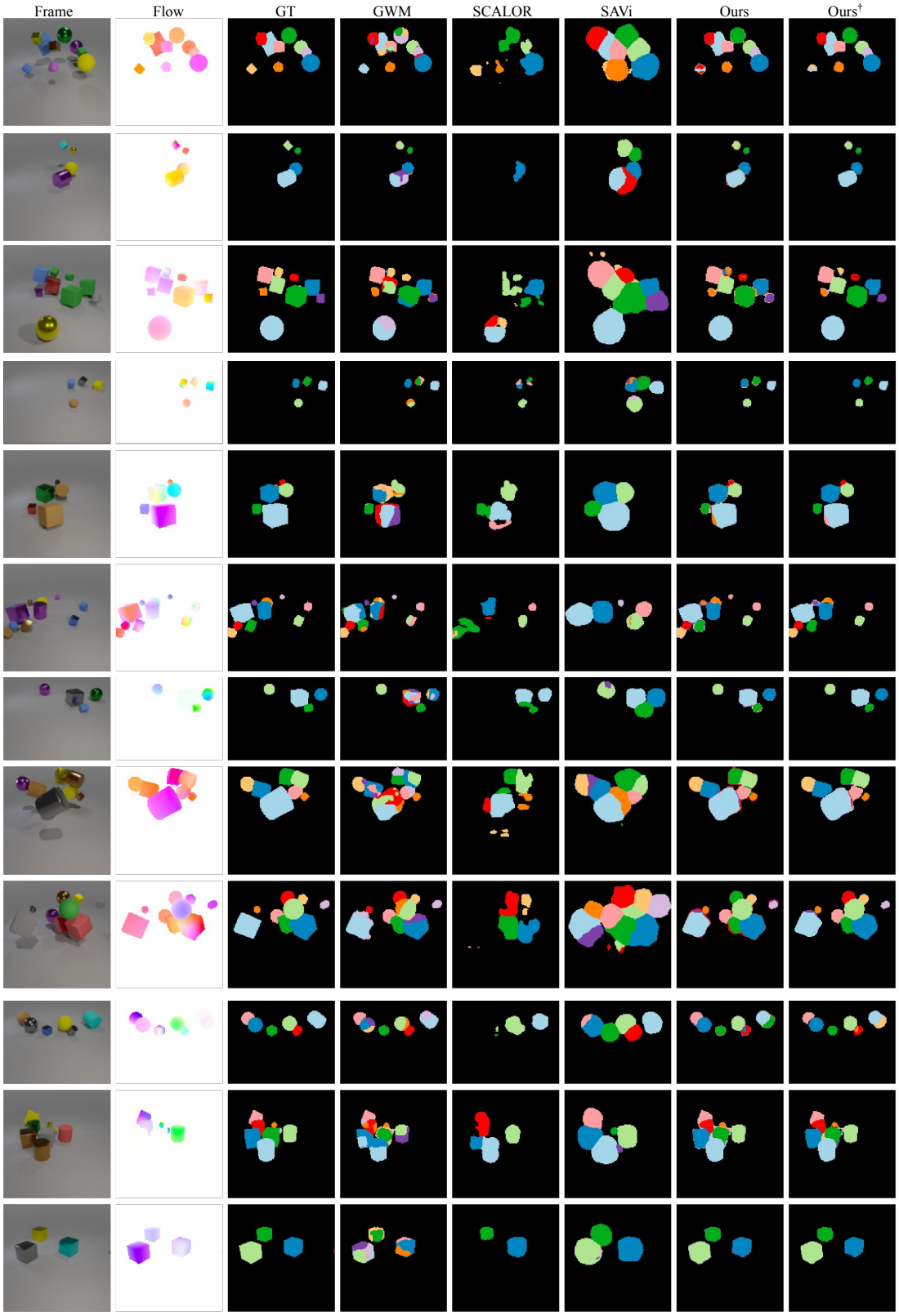


Figure D.1: Additional qualitative comparison on MOVi-A. Our method performs consistently well compared to other methods. †– indicates post-processing.

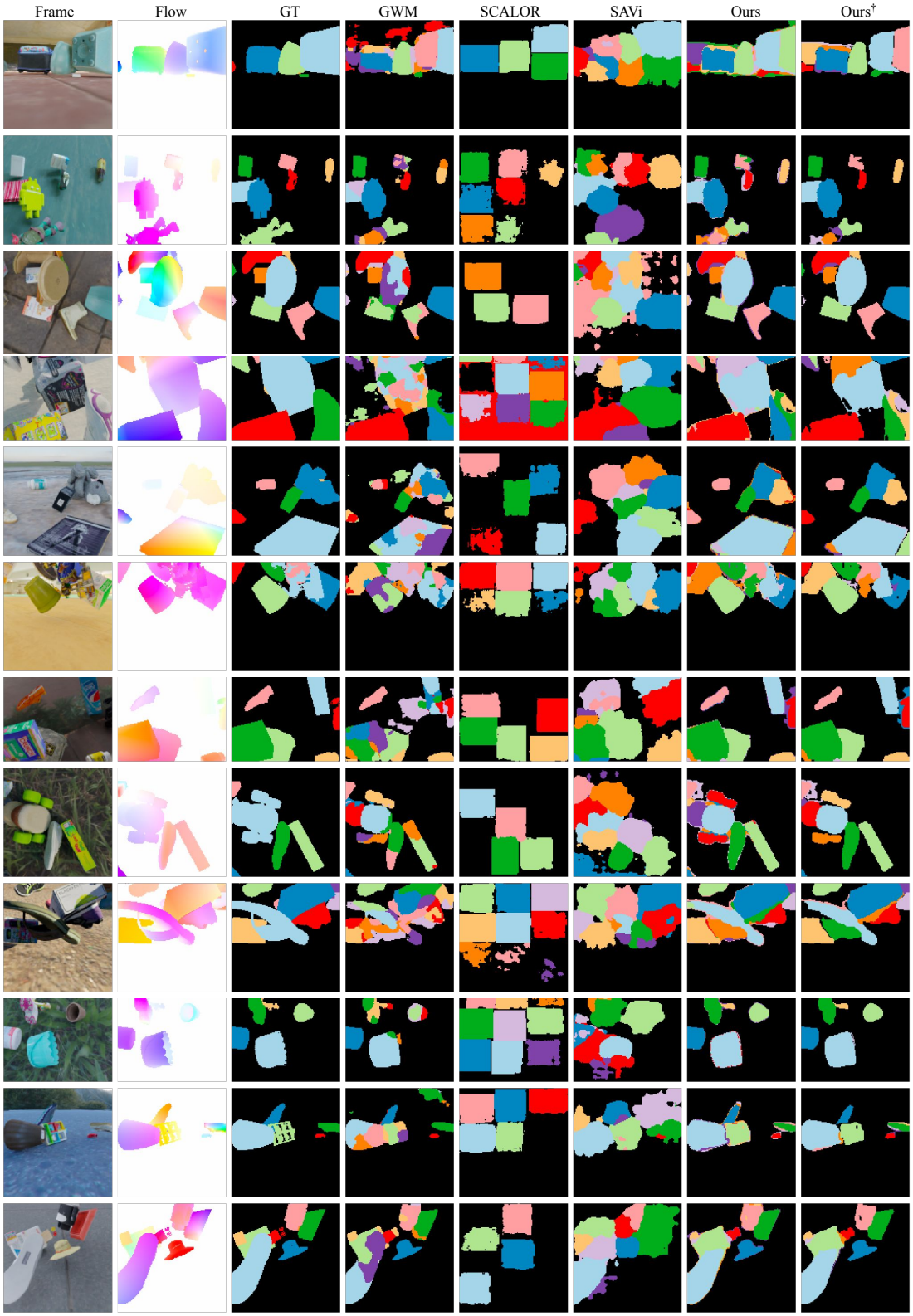


Figure D.2: Additional qualitative comparison on MOVi-C. †– indicates post-processing.

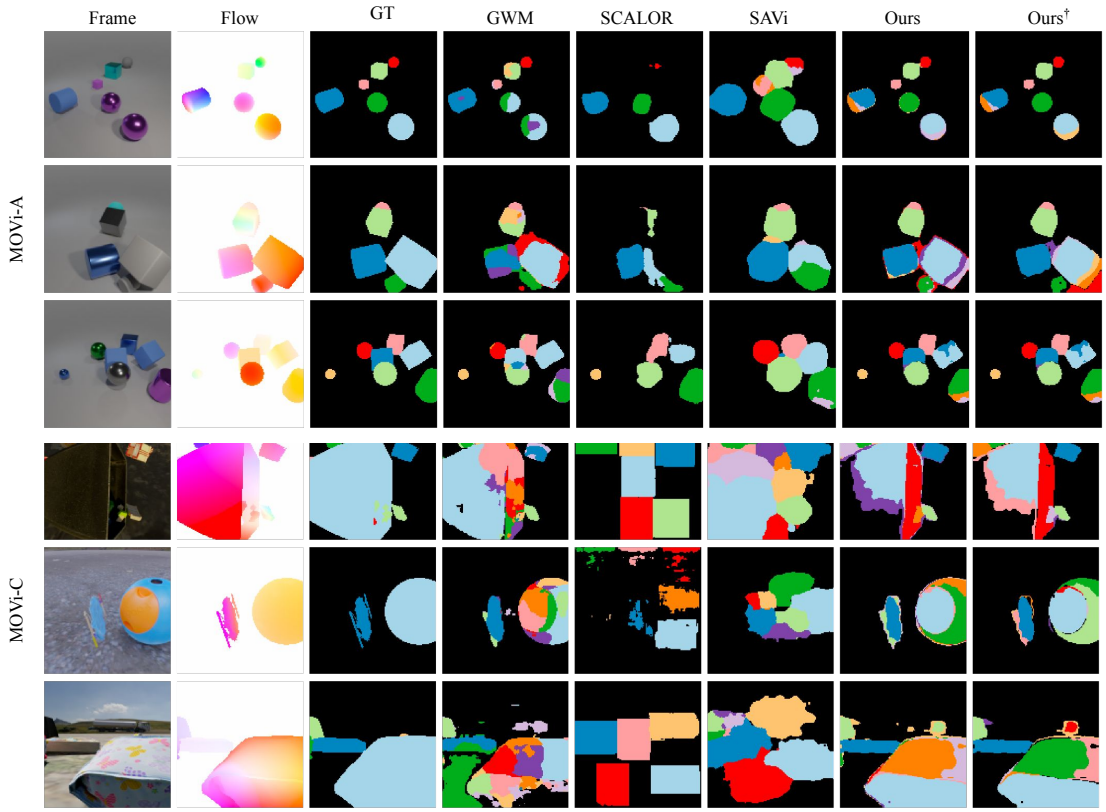


Figure D.3: Additional examples of failure cases on MOVi-A and MOVi-C. †– indicates post-processing.

Appendix E

Learning segmentation from point trajectories Supplementary Material

In this supplementary material, we consider additional ablations of our approach (appendix E.2), include further results (appendix E.3), and provide the implementation details (appendix E.5). Accompanying this supplementary material, we include videos of our results on DAVIS and SegTrackv2 datasets. We also include an example video visualising a sample of trajectories that the model receives as input. The code and models will be released upon acceptance.

E.1 Broader impact

Segmentation is a component in a very large and diverse spectrum of applications in healthcare, image processing, computer graphics, surveillance and more. As with many technologies, the application can be good or bad. In this paper, we explore how to train a model to perform segmentation in an unsupervised manner. This has the positive benefit of removing manual labour requirements to obtain annotations, which might also eventually apply to bad actors. We, however, consider the immediate real-world impact beyond the research community of our work here limited as unsupervised systems still show lower performance than supervised counterparts.

Table E.1: **Influence of** r, the rank of the trajectory matrix used in loss function (5.4).

r	**DAVIS** ($\mathcal{J} \uparrow$)
3	76.0
4	79.6
5	82.2
6	80.9

Table E.2: **Influence of** k, the number of predicted components before merging.

k	**DAVIS** $\mathcal{J} \uparrow$
2	78.0
3	82.0
4	82.2
5	72.8

Table E.3: **Influence of context length** of the trajectory matrix.

Context length	**DAVIS** ($\mathcal{J} \uparrow$)
10	79.1
15	81.0
20	82.2
30	80.8

Table E.4: **Influence of trackers** used to estimate point trajectories.

Tracker	**DAVIS** ($\mathcal{J} \uparrow$)
TAPIR [Doersch et al. 2023]	73.4
PIPs++ [Zheng et al. 2023]	74.9
BootsTap [Doersch et al. 2024]	76.8
CoTracker [Karaev et al. 2024]	78.9

E.2 Additional ablations

Rank r of trajectory matrix. In table E.1, we vary r, the rank of the trajectory matrix used in the trajectory loss (eq. (5.4)). As previously mentioned, the choice of rank reflects the degrees of freedom in the system and controls implicitly the assumptions about the types of motion and cameras used to capture sequences. At $r = 3$ and 4, we observe slight impact on the performance in comparison to $r = 5$. $r = 5$ appears to be the optimal setting, which is what we used in our main experiments. At $r = 6$, the performance drops again, likely as it becomes sufficient to group and explain simple motions together.

Number of segments k. In table E.2, we vary k, the number of masks predicted by our method, before merging. As in prior work [Choudhury et al. 2022], the $k = 4$ appears to be the optimal setting. The performance drops beyond this point as it becomes difficult to group objects.

Influence of context window length T. In table E.3, we vary the length of the context windows (f) and thus T for our method when considering trajectories. We find increasing the context window helps slightly. However, the performance starts to drop afterwards. We hypothesise that this is due to difficulty predicting sensible point trajectories for points that move outside of the frame and become invisible, as DAVIS contains many videos where the camera tracks the main subject. Though several values of this setting are viable.

Source of tracks. In table E.4, we experiment with different trackers to obtain tracks. We consider TAPIR[1], PIP++[2] and BootsTap[3] along CoTracker. Due to the limitation of some options (PIPs++ not predicting visibility) and inherent noise in invisible tracks for TAPIR, we lowered the context window to 15. We also do not consider tracks from adjacent frames as this seems to lower performance for other trackers. Finally we did not use EMA in these experiments. We observe that CoTracker performs the best while other trackers show slightly weaker results. We hypothesise this is due to CoTracker estimating reasonable trajectories for occluded points, which are included in the matrix P_k. Some trackers, e.g. TAPIR, are restricted to predicting points within the frame, thus providing extremely noisy estimates in scenes where objects move outside the frame.

Alternative networks. As our proposed loss function is network-architecture agnostic as it only requires mask prediction. Thus, any network which predicts masks or has mask-like representation could be used. In table E.5, we experiment with changing the segmenter architecture in the DAVIS benchmark. This shows that we can swap different network architectures with relative ease and obtain similar results.

Inference speed. Here we provide the inference time comparison using different networks as average FPS during DAVIS evaluation. For MaskFormer + DINO configuration, we measure 3.3 FPS, while with UNet we measure 6.4 FPS. Note that since our contribution is a loss function, it is network architecture agnostic. Using it does not affect inference time; only the choice of network architecture does. We matched the architecture with prior work for the best comparisons.

Comparison with appearance-only works. Finally, we include a comparison to unsupervised methods that consider only appearance during learning. In table E.6, we provide a comparison of VideoCutLER [Xudong Wang et al. 2023b] and VideoSAUR [Zadaianchuk et al. 2024] on DAVIS using the same merging strategy for combining multiple predictions to a binary segmentation as in our method.

[1]Code and models available https://github.com/google-deepmind/tapnet under Apache-2.0 license.

[2]Code and models available https://github.com/aharley/pips2 under MIT license.

[3]Code and models available https://github.com/google-deepmind/tapnet under Apache-2.0 license.

Table E.5: **Alternative network architectures** for segmentation.

Network	**DAVIS** ($\mathcal{J}\uparrow$)
UNet	80.6
MaskFormer + Swin-Tiny	81.2
MaskFormer + DINO	82.2

Table E.6: **Comparison with appearance-only methods.**

Method	**DAVIS** ($\mathcal{J}\uparrow$)
VideoCutLER [Xudong Wang et al. 2023b]	67.2
VideoSAUR [Zadaianchuk et al. 2024]	17.5
LRTL (Ours)	82.2

Our method shows a significant advantage. We observe that VideoCutLER has trouble segmenting instances from crowds in the background. VideoSAUR has imprecise object boundaries which severely impacts performance when measurred using Jaccard score.

E.3 Additional results

E.3.1 Qualitative results on SegTrackv2

In fig. E.1, we provide additional qualitative results from our approach on the SegTrackv2 dataset. We compare with the state-of-the-art multi-stage Relaxed Common Fate (RCF) approach [Lian et al. 2023]. Our method correctly identifies more parts of the objects and has better boundaries.

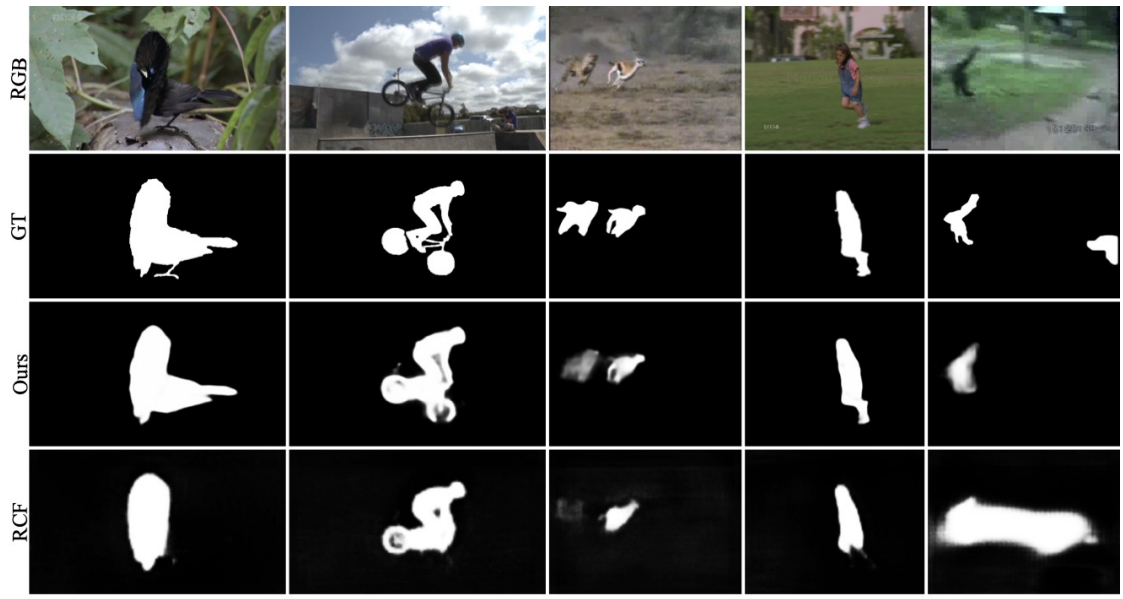


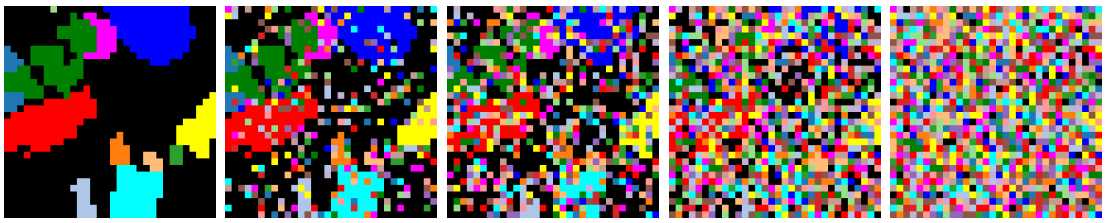
Figure E.1: Qualitative comparison of our results on SegTrackv2 with RCF which uses higher resolution and multi-stage training. Our method contains slightly better boundaries and segments more whole objects.

E.4 Parametric mask alterations

In this section, we show the effect of the parametric ground truth mask alterations used to study the trajectory loss in section 4.1. The purpose of these alterations is to disturb ground truth masks in a controlled way to enable studying the effect this has on the loss. For this purpose, we use synthetic data from MOVi-F sequences of the Kubric [Greff et al. 2022] dataset suite, which is the same data that is used to train CoTracker [Karaev et al. 2024], TAPIR [Doersch et al. 2023], PIP [Harley et al. 2022] and similar. We consider three types of alterations:

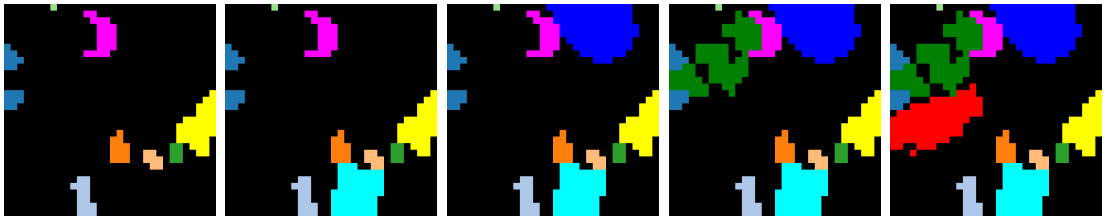
- The first kind of alteration is random *noise*. With probability η, we set each mask pixel to a random class sampled from $\mathcal{U}(0, K)$, where $K = 20$ in this case. When $\eta = 0$, thus, there is no alteration. When $\eta = 0.5$, around half of the mask pixels (in expectation) are assigned randomly. fig. E.2 shows the effect of η in practice.

- The second kind of alteration we consider is a *structural* change meant to approximate over/under-segmentation. For under-segmentation, we change the mask regions corresponding to the whole object to the background. fig. E.3 shows this in effect. For over-segmentation, we split an existing component randomly along an axis passing through the object centre and parallel to either the x- or y-axis at random. fig. E.4 shows this in effect. We parameterise



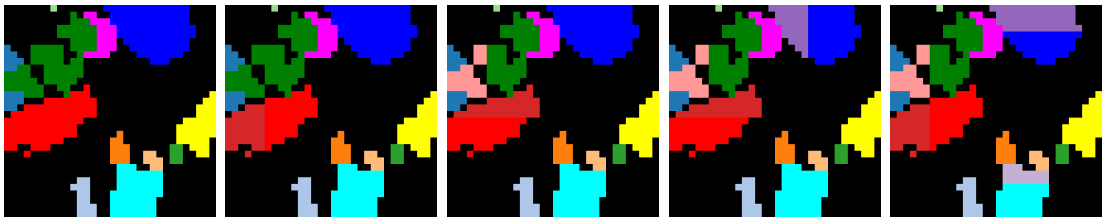
$\eta = 0.0$ $\eta = 0.25$ $\eta = 0.5$ $\eta = 0.75$ $\eta = 1.0$

Figure E.2: Example *noise* mask alteration. The parameter η is the probability of assigning a mask pixel at random.



$s = -4$ $s = -3$ $s = -2$ $s = -15$ $s = 0$

Figure E.3: Example mask *structural* mask alteration showing modification used to approximate under-segmentation. The parameter s controls the number of objects set to the background.



$s = 0$ $s = 1$ $s = 2$ $s = 3$ $s = 4$

Figure E.4: Example *structural* mask alteration modelling over-segmentation. The parameter s controls the number if objects split into two at random.

this alteration with integers s, where a positive number controls the number of components split, and negative numbers correspond to the number of components set to the background.

- The third kind of alteration is *temperature*. Its purpose is to model how the entropy of the categorical distribution modelled by the segmentation network might affect the loss. For this, we increase the temperature τ in softmax operation $softmax(l/\tau)$ for logits l calculated from the input mask, which results in increasingly "soft" masks.

The three types of alteration are composed to generate a synthetic prediction mask that can be used to investigate how the trajectory loss behaves as the mask changes.

We use 25 trials to estimate the value of the loss for a given configuration of the parameters s, η, τ.

E.5 Implementation details

Here, we further specify the configuration and implementation details used in our experiments.

E.5.1 Extracting flow

We use RAFT [Teed and J. Deng 2020][4] to extract optical flow pretrained on FlyingThings3D [Mayer et al. 2016]. We follow the methods used to extract flow in previous work [Choudhury et al. 2022; Charig Yang et al. 2021]. Namely, we consider pairs of frames with a distance in time of either 1 or 2, both in forward and backward directions for DAVIS and SegTrackv2. For FBMS, we consider distances of 3 and 6 due to lower motion setting in the dataset. The optical flow is extracted before training.

E.5.2 Extracting trajectories

We use CoTracker [Karaev et al. 2024] to extract point trajectories. CoTracker is trained on MOVi-F Kubric [Greff et al. 2022] datasets. We use CoTracker v2[5]. We query at every 4th-pixel coordinate for each frame to extract point trajectories. At 480×854 resolution for DAVIS, this results in 25k points for each frame. When tracking, we find it beneficial to inject auxiliary query points. For this, we define two additional query grids with a stride of 32, querying a frame seven frames in the future and the past (or less if at the video boundaries). This generates around 2k additional points, which we do not use for training. When processing videos of heterogeneous resolutions, we resize the input to 480×854 to maintain the same number of points.

[4]Code and models available https://github.com/princeton-vl/RAFT/tree/master under BSD-3 license.

[5]Code and models available https://github.com/facebookresearch/co-tracker under non-comercial license.

E.5.3 Training hyperparameters

For the segmentation network, we use the same model architecture as in [Choudhury et al. 2022] – MaskFormer with DINO backbone. We feed images at 192×352 resolution. We also use random horizontal flipping augmentation. The network is trained to predict $k = 4$ components, which, in the case of binary segmentation, are then merged into two following [Choudhury et al. 2022]. We train using AdamW optimiser, with a learning rate of 1.5e-4, weight decay of 0.01, a batch size of 8, and a linear learning warmup schedule for 1500 iterations. We train for 5000 iterations.[6] We use an Exponential Moving Average (EMA) with the decay power of 2/3 with a warmup of 1500 iterations and update every 10 steps to help stabilise the training. On SegTrackv2 we instead used decay power 4/5 as the dataset is considerably smaller than others. We set $\lambda_f = 0.03$, $\lambda_t = 5 * 10^-5$, and $\lambda_\tau = 0.1$ in all experiments, which yields loss values in a similar numerical range. For the temporal smoothing loss, we use $\Delta t = 5$.

E.5.4 MOVi-F experiments

When conducting experiments on the MOVi-F dataset (Sec. 4.2), we consider ground-truth trajectories obtained from modified rendering script [Greff et al. 2022]. We normalise the trajectories to the $[0, 1]$ range based on image width and height.

For K-Means, we consider the trajectories with the initial position at $T = 0$ subtracted, thus clustering offsets from the initial position.

For SSC [Elhamifar and Vidal 2013], we translate the method to Python following the original implementation in Matlab[7]. We use the ADMM variant, which we found to give better results. We set the $\alpha = 100$ and kept the rest of the hyperparameters unchanged. To transform the coefficient matrix into a graph adjacency, we found that simple symmetrisation yielded slightly better results than the proposed method that additionally normalised and filtered values. We report results for this method using the optimal number of clusters for spectral clustering.

For LRR [C.-Y. Lu et al. 2012], we similarly translate the method to Python

[6] We estimate about 3 hours to train a model using A6000 GPU (peak GPU memory 25GB). We estimate around 100 GPU hours to train models for the results here.

[7] Code available at http://www.vision.jhu.edu/code/

following the original implementation in Matlab[8]. We use $\lambda = 0.2$. Additionally, we found it beneficial to reduce $\rho = 1.01$ and use a larger number of iterations (10k) than proposed. Similarly to SSC, we experimented with different ways to transform the coefficient matrix to adjacency, including automatically determining the number of clusters based on the block-diagonal structure. We found, however, that using simpler symmetrisation with optimal numbers of clusters determined by an oracle gave the best results.

When considering our trajectory loss, we parameterise the masks with a small randomly initialised Unet [Ronneberger et al. 2015] predicting a 25-way segmentation, which we optimize using AdamW optimizer.

Note that K-Means, SSC and LRR baselines cluster trajectories rather than segmenting the image. To map back to the image domain and obtain segmentation masks, we repeatedly apply the method for each frame within a sequence, considering the trajectory for each pixel. This enables the most direct way to establish segmentation of the images through significant additional computation effort. An alternative could be to consider sequence wide-trajectories jointly; however, approaches like SSC and LLR do not scale well to such a large number of trajectories. For our trajectory loss, optimisation can be performed per sequence and, as we show in our real-world experiments, dataset-wide.

[8]Code available at https://sites.google.com/site/guangcanliu/

Appendix F

Diffusion Models for Open-Vocabulary Segmentation Supplementary Material

In this supplementary material, we provide additional experimental results, including further ablations and qualitative comparisons (appendix F.1), consider the limitations and broader impacts of our work (appendix F.2), and conclude with additional details concerning the implementation (appendix F.3).

F.1 Additional experiments

This section provides additional experimental results of OVDiff.

F.1.1 Additional Comparisons

Category filter. To ensure that the category pre-filtering does not give our approach an unfair advantage, we augment two methods (TCL [Cha et al. 2023] and OVSegmentor [Jilan Xu et al. 2023], which are the closest baselines with code and checkpoints available) with our category pre-filtering. We evaluate on the Pascal VOC dataset (where the category filter shows a significant impact; see Table 3) and report the results in table F.2. We observe that TCL improves by 0.6, while the performance of OVSegmentor drops by 0.1. On the contrary, our method benefits substantially from this component, but it still shows stronger performance

without the filter than baselines with.

Influence of Γ segmentation method. We also further investigate the use of CutLER [Xudong Wang et al. 2023a] to obtain segmentation masks. We also provide example results of segmentation in fig. F.4. In table F.3, we devise a baseline where CutLER-predicted masks are used to average the CLIP image encoder's final spatial tokens after projection. Averaged tokens are compared with CLIP text embeddings to assign a class. While relying on pre-trained components (like ours), this avoids support set generation. In the same table, we also consider whether the objectness prior provided by CutLER could be beneficial to other methods as well. We consider a version of TCL [Cha et al. 2023] and OVSegmentor [Jilan Xu et al. 2023] which we augment with CutLER. That is, after methods assign class probabilities to each pixel/patch, a majority voting for a class is performed in every region predicted by CutLER. This combines CutLER's understanding of objects and their boundaries, aspects where prior methods struggle, with open-vocabulary segmentation. However, we observe that this negatively impacts the performance of these methods, which we attribute to only a limited performance of CutLER in complex scenes present in the datasets. Finally, we also include a version of OVDiff that does not rely on CutLER for mask extractions, instead using thresholded masks. We observe that such a version of our method also has strong performance.

We additionally experiment with stronger segmenters to understand the influence of FG/BG mask quality. We replace our FG/BG segmentation approach with strong supervised models: with SAM, we achieve 67.1 on VOC, and with Grounded SAM, 68.5. This slightly improves results from 66.3 of our configuration with CutLER, but the performance gain is not large and thus not critical.

Influence of image generator. We experiment with different SD versions in table F.1 and observe improvement with more advanced generators.

Table F.1: Influence of different text-to-image generators.

T2I	VOC
SD 1.5	66.4
SD 2.0	67.7
SD 2.1	67.1
Hyper-SD	67.7

Class prompts. We additionally consider whether corrections introduced to class prompts might have similarly provided additional benefits to our approach (see appendix F.3.3 for details). To that end, we also evaluate TCL and OVSeg-

Table F.2: Use of category filter component. OVDiff without category filter outperforms prior work with cat. filter.

Model	Category filter ✗	Category filter ✓
OVSegmentor	53.8	53.7
TCL	51.2	51.8
TCL (+PAMR)	55.0	56.0
OVDiff	**56.2**	**66.4**

Table F.3: Application of CutLER. Prior work does not benefit from using CutLER during inference, while OVDiff shows strong results without it.

Model	CutLER	VOC	Context	Object
CLIP	✓	33.0	11.6	11.1
OVSegmentor		53.8	20.4	25.1
OVSegmentor	✓	38.7	14.4	16.8
TCL		51.2	24.3	30.4
TCL	✓	43.1	20.5	22.7
OVDiff		62.8	28.6	34.9
OVDiff	✓	**66.3 ± 0.2**	**29.7 ± 0.3**	**34.6 ± 0.3**

menter (methods that do not rely on additional prompt curation) with our corrected prompts and consider a version of our method without such corrections in table F.4. We observe only marginal to no impact on the performance.

Prompt template Finally, we consider the prompt template employed when sampling support image set: "`A good picture of a` $\langle c_i \rangle$" for class prompt c_i. This template is generic and broadly applicable to virtually any natural language specification of a target class. While prior work adopts prompt expansion by considering a list of synonyms and subcategories, it is not entirely clear how such a strategy could be systematically performed for any in-the-wild prompts, such as a "chocolate glazed donut". We experiment with a list of synonyms and subclasses, as employed by [Ranasinghe et al. 2023], on VOC datasets measuring 66.4 mIoU, which is similar to our single prompt performance 66.3 ± 0.2. Curating such lists automatically is an interesting future scaling direction.

F.1.2 Additional ablations

Prototype combinations. In table F.7, we consider the three different types of prototypes described in Section 3 and test their performance individually and in various combinations. We find that the "part" prototypes obtained by K-

Table F.4: Using corrected prompts. We consider if corrected class names benefit prior work. We observe negligible to no effect.

Model	Correction	VOC	Context	Object
OVSegmentor		53.8	20.4	25.1
OVSegmentor	✓	53.9	20.4	25.1
TCL		51.2	24.3	30.4
TCL	✓	50.6	24.3	30.4
OVDiff		66.1	29.5	34.9
OVDiff	✓	**66.3 ± 0.2**	**29.7 ± 0.3**	**34.6 ± 0.3**

Table F.5: Choice of K for number of centroids.

K	VOC	Context
8	63.8	29.2
16	64.0	29.3
32	64.4	29.4
64	64.3	28.0

Table F.6: Ablation of different SD feature configurations. Removing first and last cross attention *layers*, mid, 1st and 2nd upsampling *blocks* (all layers in the block) has a negative effect.

1st layer	Mid block	Up-1 block	Up-2 block	Last layer	Context
✓	✓	✓	✓	✓	29.4
	✓	✓	✓	✓	29.4
✓		✓	✓	✓	29.2
✓	✓		✓	✓	27.3
✓	✓	✓		✓	28.9
✓	✓	✓	✓		29.3

means clustering show strong performance when considered individually on VOC. Instance prototypes show strong individual performance on Context, as well as in combination with the average category prototype. The combination of all three types shows the strongest results across the two datasets, which is what we adopt in our main set of experiments.

We also consider the treatment of prototypes under the stuff filter. We investigate the impact of not excluding background prototypes for “stuff" classes. In this setting, we measure 29.1 on Context, which is a slight reduction in performance. We also investigate the benefit of categorisation into “things” and “stuff” used in the stuff filter component. Instead, we filter all background prototypes using all foreground prototypes. In this configuration, we measure 27.6 on Context. Both configurations show a reduction from 29.4, measuring using the stuff filter with

Table F.7: Ablation of various configurations for prototypes. We consider average $\bar{P}$, instance P_n, and part P_k prototypes individually and in various combinations on VOC and Context datasets. Combination of all three types of prototypes shows strongest results.

$\bar{\mathbf{P}}$	$\mathbf{P_n}$	$\mathbf{P_k}$	VOC	Context
✓	✓	✓	64.4	29.4
✓		✓	61.7	29.3
✓	✓		63.5	29.4
	✓	✓	62.5	28.4
		✓	63.7	28.8
	✓		60.0	29.0
✓			62.5	28.4

Figure F.1: Qualitative comparison on in-the-wild images. OVDiff performs significantly better than prior state-of-the-art, TCL, on wildlife images containing multiple instances, studio photos with simple backgrounds, images containing multiple categories and an image containing a rare instance of a class.

categorisation in "stuff" and "things", as used in our main experiments. Finally, we experiment by removing part-level prototypes for "stuff" classes, which also results in a performance drop to 28.0.

K - number of clusters. In table F.5, we investigate the sensitivity of the method to the choice of K for the number of "part" prototypes extracted using K-means clustering. Although our setting $K = 32$ obtains slightly better results on Context and VOC, other values result in comparable segmentation performance suggesting that OVDiff is not sensitive to the choice of K and a range of values is viable.

SD features. When using Stable Diffusion as a feature extractor, we consider various combinations of layers/blocks in the UNet architecture. We follow the nomenclature used in the Stable Diffusion implementation where consecutive layers of Unet are organised into *blocks*. There are 3 down-sampling blocks with 2 cross-attention layers each, a mid-block with a single cross-attention, and 3 up-sampling blocks with 3 cross-attention layers each. We report our findings in table F.6. Including the first and last cross-attention layers in the feature extraction process has a small positive impact on segmentation performance, which we attribute to the high feature resolution. We also consider excluding features from the middle block of the network due to small 8×8 resolution but observe a small negative impact on performance on the Context dataset. We also investigate whether including the first (Up-1) and the second upsampling (Up-2) blocks are necessary. Without them, the performance drops the most out of the configurations considered. Thus, we use a concatenation of features from the middle, first and second upsampling blocks and the first and last layers in our main experiments.

F.1.3 Qualitative results

We include additional qualitative results from the benchmark datasets in fig. F.2. Our method achieves high-quality segmentation across all examples without any post-processing or refinement steps. In fig. F.3, we show examples of support images sampled for some things, and stuff categories. In fig. F.5, we show examples of support set images sampled for rare *pikachu* class.

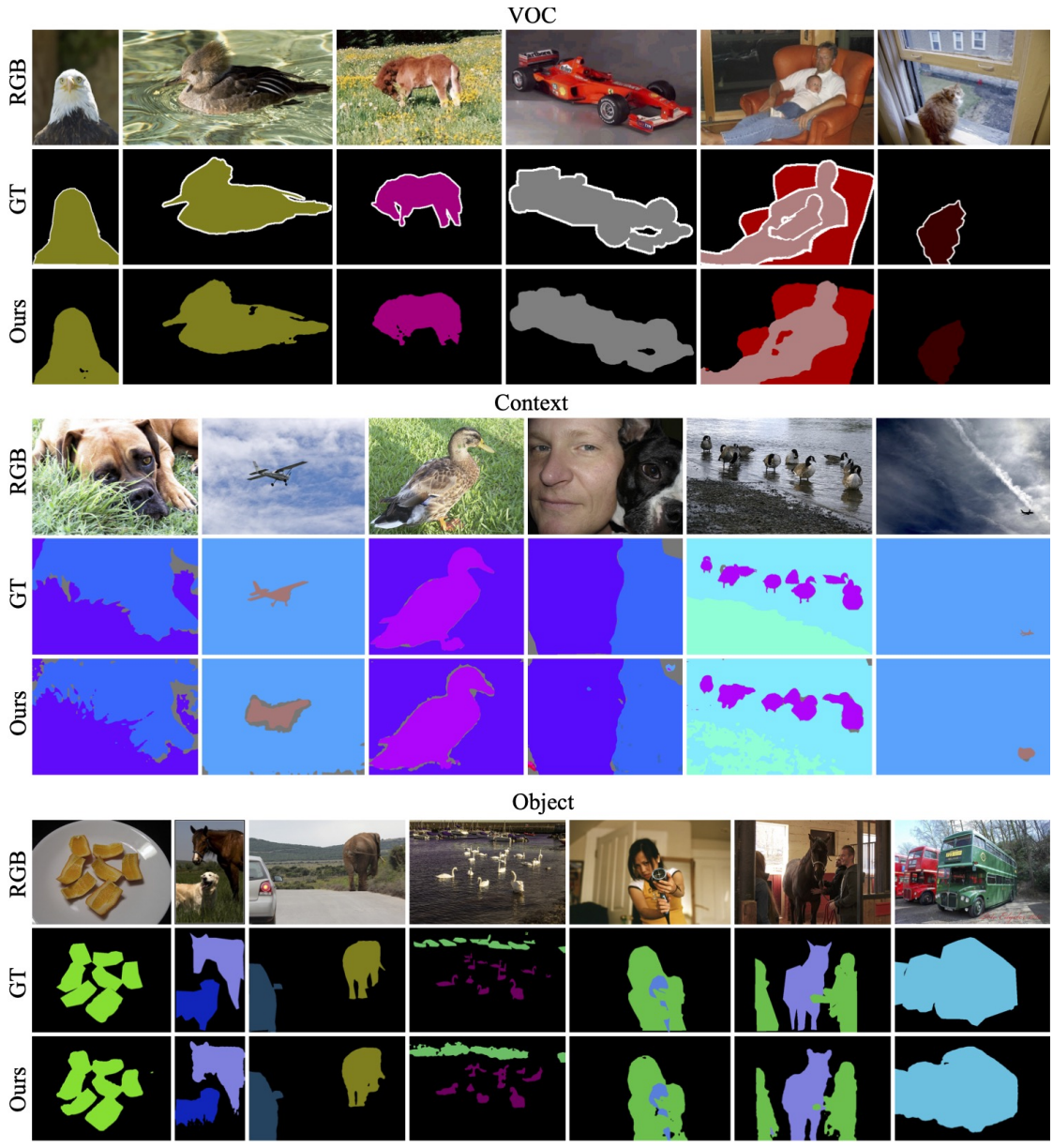
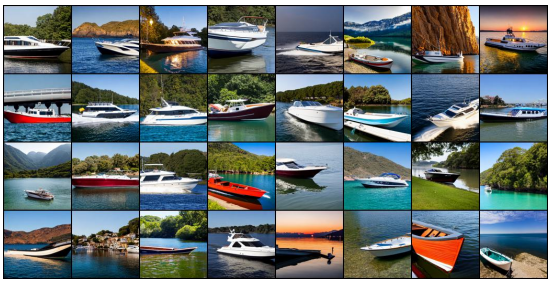


Figure F.2: Additional qualitative results. Images from Pascal VOC (top), Pascal Context (middle), and COCO Object (bottom).



(a) boat

(b) person

(c) sky

(d) water

(e) light

(f) parking meter

(g) mountain

(h) horse

Figure F.3: Images sampled for a support set of some categories.

F.2 Broader impact

Semantic segmentation is a component in a vast and diverse spectrum of applications in healthcare, image processing, computer graphics, surveillance and more. As for any foundational technology, applications can be good or bad. OVDiff is similarly widely applicable. It also makes it easier to use semantic segmentation in new applications by leveraging existing and new pre-trained models. This is a bonus for inclusivity, affordability, and, potentially, environmental impact (as it requires no additional training, which is usually computationally intensive); however, these features also mean that it is easier for bad actors to use the technology.

Because OVDiff does not require further training, it is more versatile but also inherits the weaknesses of the components it is built on. For example, it might contain the biases (e.g., gender bias) of its components, in particular Stable Diffusion [Schramowski et al. 2023], which is used for generating support images for any given category/description. Thus, it should not be exposed without further filtering and detection of, e.g., NSFW material in the sampled support set. Finally, OVDiff is also bound by the licenses of its components.

F.2.1 Limitations

As OVDiff relies on pretrained components, it inherits some of their limitations. OVDiff works with the limited resolution of feature extractors, due to which it might occasionally miss tiny objects. Furthermore, OVDiff cannot segment what the generator cannot generate. For example, current diffusion models struggle with producing legible text, which can make it difficult to segment specific words. Furthermore, applications in domains far from the generator's training data (e.g. medical imaging) are unlikely to work out of the box.

F.3 OVDiff: Further details

In this section, we provide additional details concerning the implementation of OVDiff. We begin with a brief overview of the attention mechanism and diffusion models central to extracting features and sampling images. We review different feature extractors used. We specify the hyperparameter setting for all our experiments and provide an overview of the exchange with ChatGPT used to categorise

classes into “thing” and “stuff”.

F.3.1 Preliminaries

Attention. In this work, we make use of pre-trained ViT [Alexey Dosovitskiy et al. 2021] networks as feature extractors, which repeatedly apply multi-headed attention layers. In an attention layer, input sequences $X \in \mathbb{R}^{l_x \times d}$ and $Y \in \mathbb{R}^{l_y \times d}$ are linearly project to forms *keys*, *queries*, and *values*: $K = W_k Y,\ Q = W_q X,\ V = W_v X$. In self-attention, $X = Y$. Attention is calculated as $A = \text{softmax}(\frac{1}{\sqrt{d}} Q K^\top)$, and softmax is applied along the sequence dimension l_y. The layer outputs an update $Z = X + A \cdot V$. ViTs use multiple heads, replicating the above process in parallel with different projection matrices W_k, W_q, W_v. In this work, we consider *queries* and *keys* of attention layers as points where useful features that form meaningful inner products can be extracted. As we detail later (Appendix F.3.2), we use the *keys* from attention layers of ViT feature extractors (DINO/MAE/CLIP), concatenating multiple heads if present.

Text-to-image diffusion models. Diffusion models are a class of generative models that form samples starting with noise and gradually denoising it. We focus on latent diffusion models [Rombach et al. 2022] which operate in the latent space of an image VAE [Kingma and Welling 2014] forming powerful conditional image generators. During training, an image is encoded into VAE latent space, forming a latent vector z_0. A noise is injected forming a sample $z_\tau \sim \mathcal{N}(z_\tau; \sqrt{1-\alpha_\tau} z_0, \alpha_\tau I)$ for timestep $\tau \in \{1 \dots T\}$, where α_τ are variance values that define a noise schedule such that the resulting z_T is approximately unit normal. A conditional UNet [Ronneberger et al. 2015], $\epsilon_\theta(z_t, t, c)$, is trained to predict the injected noise, minimising the mean squared error $\mathbb{E}_t \left(\alpha_t \| \epsilon_\theta(z_t, t, c) - z_0 \|_2\right)$ for some caption c and additional constants a_t. The network forms new samples by reversing the noise-injecting chain. Starting from $\hat{z}_T \sim \mathcal{N}(\hat{z}_T; 0, I)$, one iterates $\hat{z}_{t-1} = \frac{1}{\sqrt{1-\alpha_t}}(\hat{z}_t + \alpha_t \epsilon_\theta(\hat{z}_t, t, c)) + \sqrt{\alpha_t}\hat{z}_t$ until $\hat{z}_0$ is formed and decoded into image space using the VAE decoder. The conditional UNet uses cross-attention layers between image patches and language (CLIP) embeddings to condition on text c and achieve text-to-image generation.

F.3.2 Feature extractors

OVDiff is buildable on top of any pre-trained feature extractor. In our experiments, we have considered several networks as feature extractors with various self-supervised training regimes:

- **DINO** [Caron et al. 2021] is a self-supervised method that trains networks by exploring alignment between multiple views using an exponential moving average teacher network. We use the ViT-B/8 model pre-trained on ImageNet[1] and extract features from the *keys* of the last attention layer.

- **MAE** [K. He et al. 2017] is a self-supervised method that uses masked image inpainting as a learning objective, where a portion of image patches are dropped, and the network seeks to reconstruct the full input. We use the ViT-L/16 model pre-trained on ImageNet at a resolution of 448 [R. Hu et al. 2022].[2] The *keys* of the last layer of the *encoder* network are used. No masking is performed.

- **CLIP** [Radford et al. 2021] is trained using image-text pairs on an internal dataset WIT-400M. We use ViT-B/16 model[3]. We consider two locations to obtain dense features: *keys* from a self-attention layer of the image encoder and *tokens* which are the outputs of transformer layers. We find that *keys* of the second-to-last layer give better performance.

- We also consider **Stable Diffusion**[4] (v1.5) itself as a feature extractor. To that end, we use the *queries* from the cross-attention layers in the UNet denoiser, which correspond to the image modality. Its UNet is organised into three downsampling blocks, a middle block, and three upsampling blocks. We observe that the middle layers have the most semantic content, so we consider the middle block, 1st and 2nd upsampling blocks and aggregate features from all three cross-attention layers in each block. As the features are quite low in resolution, we include the first downsampling cross-attention layer and the last upsampling cross-attention layer as well. The feature maps are bilinearly upsampled to resolution 64×64 and concatenated. A noise appropriate for

[1]Model and code available at https://github.com/facebookresearch/dino.
[2]Model and code from https://github.com/facebookresearch/long_seq_mae.
[3]Model and code from https://github.com/openai/CLIP.
[4]We use implementation from https://github.com/huggingface/diffusers.

Figure F.4: FG/BG segmentation of classes of *water*, *snow* and *grass*. The foreground is in red, while the background is shown in blue.

Figure F.5: Example images from the support set of a rare *pikachu* class.

$\tau = 200$ timesteps is added to the input. For feature extraction, we run SD in *unconditional* mode, supplying an empty string for text caption.

F.3.3 Datasets

We evaluate on validation splits of PASCAL VOC (VOC), Pascal Context (Context) and COCO-Object (Object) datasets. PASCAL VOC [Mark Everingham et al. 2010; M. Everingham et al. 2012] has 21 classes: 20 foreground plus a background class. For Pascal Context [Mottaghi et al. 2014], we use the common variant with 59 foreground classes and 1 background class. It contains both "things" and "stuff" classes. The COCO-Object is a variant of COCO-Stuff [Caesar et al. 2018] with 80 "thing" classes and one class for the background. Textual class names are used as natural language specifications of names. We renamed or specified certain class names to fix errors (e.g. `pottedplant` → `potted plant`), resolve ambiguity better (e.g. `mouse` → `computer mouse`) or change to more common spelling/word (e.g. `aeroplane` → `airplane`), resulting in 14 fixes. We experiment and measure the impact of this in appendix F.1.1 for our and prior work.

F.3.4 Comparative baselines

We briefly review the prior work in used in our experiments, mainly in Table 1. We consider baselines that do not rely on mask annotations and have code and checkpoints available or detail their evaluation protocol that matches that used in other prior works [Jiarui Xu et al. 2022; Cha et al. 2023; Jilan Xu et al. 2023].Most prior work [Q. Liu et al. 2022; Cha et al. 2023; Jiarui Xu et al. 2022; Ren et al. 2023; H. Luo et al. 2023; Jilan Xu et al. 2023] trains image and text encoders on

large image-text datasets with a contrastive loss. The methods mainly differ in their architecture and use of grouping mechanisms to ground image-level text on regions. ViL-Seg [Q. Liu et al. 2022] uses online clustering, GroupViT [Jiarui Xu et al. 2022] and ViewCo [Ren et al. 2023] employ group tokens. OVSegmentor [Jilan Xu et al. 2023] uses slot-attention and SegCLIP [H. Luo et al. 2023] a grouping mechanism with learnable centers. CLIPPy [Ranasinghe et al. 2023], TCL [Cha et al. 2023], and MaskCLIP [C. Zhou et al. 2022] predict classes for each image patch: [Ranasinghe et al. 2023] use max-pooling aggregation, [Cha et al. 2023] self-masking, and [C. Zhou et al. 2022] modify CLIP for dense predictions. To assign a background label [Q. Liu et al. 2022; Cha et al. 2023; Jiarui Xu et al. 2022; Ren et al. 2023; H. Luo et al. 2023] use thresholding while [Ranasinghe et al. 2023] uses dataset-specific prompts. CLIP-DIY [Wysoczańska et al. 2024] leverages CLIP as a zero-shot classifier and applies it on multiple scales to form a dense segmentation. ReCO [Shin et al. 2022b] is closer in spirit to our approach as it uses a support set for each prompt; this set, however, is CLIP-retrieved from curated image collections, which may not be applicable for any category in-the-wild. The conceptual difference between OVDiff and ReCO is that OVDiff emphasises and preserves *diverse* prototypes by construction: generation overcomes a limited database; sampled images are segmented individually preserving unique visuals of each instance rather than co-segmenting, which leverages commonality. We construct multiple prototypes at multiple levels of granularity to similar effect, as opposed to averaging in ReCO.

We also note that prior work builds on top of similar pre-trained components such as CLIP in [Shin et al. 2022b; C. Zhou et al. 2022; Cha et al. 2023; H. Luo et al. 2023], OpenCLIP in [Wysoczańska et al. 2024], DINO + T5/RoBERTa in [Ranasinghe et al. 2023; Jilan Xu et al. 2023]. We additionally make use of StableDiffusion, which is trained on a larger dataset (3B, compared to 400M of CLIP or 2B or OpenCLIP). OVDiff is, however, fundamentally different to all prior work, as (a) it generates a support set of synthetic images given a class description, and (b) it does not rely on additional training data and further training for learning to segment.

F.3.5 Hyperparameters

OVDiff has relatively few hyperparameters and we use the same set in all experiments. Unless otherwise specified, $N = 32$ images are sampled using classifier-free guidance scale [Ho and Salimans 2021] of 8.0 and 30 denoising steps. We employ `DPM-Solver` scheduler [C. Lu et al. 2022]. When sampling images for the support sets, we also use a negative prompt "*text, low quality, blurry, cartoon, meme, low resolution, bad, poor, faded*". If/when segmenter Γ fails to extract any components in a sampled image, a fallback of adaptive thresholding of A_n is used, following [Liao et al. 2001]. During inference, we set $\eta = 10$, which results in 1024 text prompts processed in parallel, a choice made mainly due to computational constraints. We set the thresholds for the "stuff" filter between background prototypes for "things" classes and the foreground of "stuff" at 0.85 for all feature extractors. When sampling, a seed is set for each category individually to aid reproducibility.

Computation cost. We focus on a construction of a method to show that existing foundational diffusion models can be used for segmentation with great efficacy without further training. OVDiff requires computing prototypes instead. With our unoptimized implementation, we measure around 110 ± 10s to calculate prototypes (sample images, extract features and aggregate) for a single category or 50.2 ± 2s without clustering using SD. Using CLIP, we measure 49.2 ± 0.2s with clustering and 47.7 ± 0.2s without. We note that sampling time grows linearly: we measure 55s for 16, 110s for 32, and 213s for 64 images per class. The prototype storage requirements are 0.39MB using CLIP/DINO for each class.

With our unoptimized implementation, we measure around 110 ± 10s to calculate prototypes using SD for a single class, or around 1.14 TFLOP/s-hours of compute. While the focus of this study is not computational efficiency, we can compare prototype sampling to the cost of additional training of other methods: TCL requires 2688, GroupViT 10752, and OVSegmentor 624 TFLOP/s-hours.[5] While training has an upfront compute cost and requires special infrastructure (e.g. OVSegmentor uses 16×A100s), OVDiff's prototype set can be grown progressively as needed, while showing better performance.

We additionally measure the speed of inference at 0.6s per image, which is slightly

[5]Estimated as training time × num. GPUs × theoretical peak TFLOP/s for GPU type.

slower but comparable to 0.2s for TCL and 0.08s for OVSegmentor. We performed inference measurements using SD on the same machine with a 2080Ti GPU using 21 classes and the same resolution/sliding window settings for all methods.

F.3.6 Interaction with ChatGPT

We interact with ChatGPT to categorise classes into “stuff” and “things” for the stuff filter component. Due to input limits, the categories are processed in blocks. Specifically, we input “*In semantic segmentation, there are "stuff" or "thing" classes. Please indicate whether the following class prompts should be considered "stuff" or "things":*”. We show the output in table F.8. Note there are several errors in the response, e.g. `glass`, `blanket`, and `trade name` are actually instances of tableware, bedding and signage, respectively, so should more appropriately be treated as “things”. Similarly, `land` and `sand` might be more appropriately handled as “stuff”, same as `snow` and `ground`. Despite this, We find ChatGPT contains sufficient knowledge when prompted with "in semantic segmentation". We have estimated the accuracy of ChatGPT in thing/stuff classification using the categories of COCO-Stuff, which are defined as 80 "things" and 91 "stuff" categories. ChatGPT achieves an accuracy rate of 88.9% in this case. We also measure the impact the potential errors have on our performance by providing “oracle" answers on the Context dataset. We measure 29.6 mIoU, which is similar to 29.7 ± 0.3 of using ChatGPT, showing that small errors do not drastically affect the method, however, enable using “stuff" filter component, which improves performance (see Table 3).

Table F.8: **Response from interaction with ChatGPT.** We used ChatGPT model to automatically categorise classes in "stuff" or "things".

airplane:	thing	window:	thing	awning:	thing
bag:	thing	wood:	stuff	streetlight:	thing
bed:	thing	windowpane:	thing	booth:	thing
bedclothes:	stuff	earth:	thing	television receiver:	thing
bench:	thing	painting:	thing	dirt track:	thing
bicycle:	thing	shelf:	thing	apparel:	thing
bird:	thing	house:	thing	pole:	thing
boat:	thing	sea:	thing	land:	thing
book:	thing	mirror:	thing	bannister:	thing
bottle:	thing	rug:	thing	escalator:	thing
building:	thing	field:	thing	ottoman:	thing
bus:	thing	armchair:	thing	buffet:	thing
cabinet:	thing	seat:	thing	poster:	thing
car:	thing	desk:	thing	stage:	thing
cat:	thing	wardrobe:	thing	van:	thing
ceiling:	stuff	lamp:	thing	ship:	thing
chair:	thing	bathtub:	thing	fountain:	thing
cloth:	stuff	railing:	thing	conveyer belt:	thing
computer:	thing	cushion:	thing	canopy:	thing
cow:	thing	base:	thing	washer:	thing
cup:	thing	box:	thing	plaything:	thing
curtain:	stuff	column:	thing	swimming pool:	thing
dog:	thing	signboard:	thing	stool:	thing
door:	thing	chest of drawers:	thing	barrel:	thing
fence:	stuff	counter:	thing	basket:	thing
floor:	stuff	sand:	thing	waterfall:	thing
flower:	thing	sink:	thing	tent:	thing
food:	thing	skyscraper:	thing	minibike:	thing
grass:	stuff	fireplace:	thing	cradle:	thing
ground:	stuff	refrigerator:	thing	oven:	thing
horse:	thing	grandstand:	thing	ball:	thing
keyboard:	thing	path:	thing	step:	stuff
light:	thing	stairs:	thing	tank:	thing
motorbike:	thing	runway:	thing	trade name:	stuff
mountain:	stuff	case:	thing	microwave:	thing
mouse:	thing	pool table:	thing	pot:	thing
person:	thing	pillow:	thing	animal:	thing
plate:	thing	screen door:	thing	lake:	stuff
platform:	stuff	stairway:	thing	dishwasher:	thing
plant:	thing	river:	thing	screen:	thing
road:	stuff	bridge:	thing	blanket:	stuff
rock:	stuff	bookcase:	thing	sculpture:	thing
sheep:	thing	blind:	thing	hood:	thing
shelves:	thing	coffee table:	thing	sconce:	thing
sidewalk:	stuff	toilet:	thing	vase:	thing
sign:	thing	hill:	thing	traffic light:	thing
sky:	stuff	countertop:	thing	tray:	stuff
snow:	stuff	stove:	thing	ashcan:	thing
sofa:	thing	palm:	thing	fan:	thing
table:	thing	kitchen island:	thing	pier:	thing
track:	stuff	swivel chair:	thing	crt screen:	thing
train:	thing	bar:	thing	bulletin board:	thing
tree:	thing	arcade machine:	thing	shower:	thing
truck:	thing	hovel:	thing	radiator:	thing
monitor:	thing	towel:	thing	glass:	stuff
wall:	stuff	tower:	thing	clock:	thing
water:	stuff	chandelier:	thing	flag:	thing