From *Video* to *Virtual*: Object-centric 3D Scene Understanding from Videos

Yash Sanjay Bhalgat

St Cross College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2025

अथातो ब्रह्मजिज्ञासा || १ ||

athāto brahmajijñāsā || 1 ||

"Now, therefore, begins the inquiry into Brahman (the ultimate reality)."

Brahma Sūtra 1.1.1

Abstract

Understanding the 3D structure of our world from casual videos remains a central challenge in computer vision. Videos provide natural supervision through motion and viewpoint changes, yet inferring geometry, objectness, and semantics directly from such unconstrained input remains difficult. This thesis develops methods for object-centric 3D scene understanding from video, combining geometric priors, neural fields, and foundation models for both static and dynamic environments.

The work begins by learning how different views relate to one another – a prerequisite for any model that aims to understand 3D structure. We teach vision transformers to internalize multi-view geometry through an epipolar-aware attention objective that softly enforces geometric consistency across viewpoints, yielding viewpoint-invariant correspondences without requiring camera poses at inference.

Once geometric reasoning is established, modeling object structure within static scenes becomes a natural next step. We achieve this by "lifting" 2D instance predictions from large segmentation models into a neural feature field with a slow-fast contrastive objective. This approach fuses inherently multi-view inconsistent information, *viz.* untracked 2D instance segmentations, to recover coherent 3D object instances in cluttered environments, without any 3D supervision.

We extend this formulation to represent not only objects but also their hierarchical and semantic relations. Our proposed nested neural feature field encodes part-object-scene structure and aligns it with language-based embeddings, enabling open-vocabulary reasoning and efficient querying of complex indoor scenes.

Finally, we add knowledge of *motion* by proposing a framework for dynamic 3D scene understanding in egocentric videos that integrates segmentation, 2D-to-3D lifting, and geometry-aware association to track objects over time, maintaining identity under motion and occlusion while supporting amodal reconstruction.

Together, these contributions unify geometry, structure, semantics, and dynamics for learning object-centric 3D representations from everyday videos.

Keywords – 3D scene understanding, multimodal, neural fields, video object tracking

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.

Yash Sanjay Bhalgat, November 2025.

Acknowledgement

Looking back at the path that led to this thesis, I am struck by how far this journey has taken me. Growing up in a small town in India with limited educational resources, the prospect of one day working with the Visual Geometry Group (VGG) at Oxford — a group at the very forefront of computer vision — seemed like a distant dream. This achievement belongs not just to me, but to the many hands that guided me when the barriers seemed insurmountable.

I owe my thanks to my DPhil supervisors – João Henriques, Andrew Zisserman, Andrea Vedaldi, and Iro Laina – for their guidance, patience, and intellectual generosity, which have shaped my growth as a researcher. Beyond their scientific advice, João's warmth and steady support often made him feel more like a friend than a supervisor; his belief in me gave me the confidence to keep going when it felt hardest. Andrew's kindness and meticulous approach to research have been deeply inspiring; his encouragement beyond technical matters was a great source of strength as I worked toward completing this thesis. Furthermore, my 4 years with the AIMS CDT program have been deeply rewarding. The program's generous funding and travel support provided the opportunity to present my work at conferences such as CVPR, NeurIPS, and ECCV. A special mention must go to Wendy Poole, whose exceptional care, efficiency, and kindness truly defined the CDT experience.

I am deeply indebted to my parents for laying the foundations during my childhood that allowed me to build the path I am on today. I am grateful for my father's incredible support, resourcefulness and foresight, and for my mother's strength, hard work, and the values she instilled in me; together, they empowered me to reach places I hadn't once imagined. My brother, Bhushan, remains one of the smartest and most hardworking people I know. He was my first mentor, teaching me to be curious and to ask the meaningful questions that defined my scientific aptitude. My sister-in-law, Ashwini, has motivated me through her hard work and dedication, and my nephew, Hridaan, has been a source of light in our lives.

This journey was also defined by my friends, who became my family away from home. I am grateful to Divyanshu Mishra, Pramit Saha, Guanqi Zhan, Shuai Chen, Felix Wagner, Xianzheng Ma, the Indian student community in Oxford, and the many others who provided laughter and intellectual companionship in equal measure. Some of them stood by me during one of the most difficult phases of my personal life, providing unwavering support and strength as I navigated my divorce. Their belief in me and their constant presence were a lifeline when I needed it most, and I am profoundly grateful for the kindness they showed me.

Throughout all these years, I am thankful to God for the grace that has followed me, often providing more than I even knew to ask for.

Lastly, as Snoop Dogg once said, I want to thank myself. I want to thank me for believing in me when things seemed impossible, for the hard work, and for the perseverance that has defined my path through life.

Contents

Chapter 1

Introduction and Background

This chapter sets out the motivation and core research questions of the thesis, together with a review of the most relevant prior work.

1.1 Motivation

We live in a three-dimensional world. Everyday videos – either from handheld smartphones or egocentric cameras – capture objects across changing viewpoints, lighting, and motion. This thesis explores using such videos to model the shape, identity, and movement of objects in 3D, potentially enabling safer robots, more useful augmented reality applications, and richer digital maps without the need for expensive 3D sensors.

Despite rapid progress in learning from large-scale unstructured data, most computer vision systems are still trained on 2D imagery and typically reason *only implicitly* about 3D structure. Understanding the world in 3D – its geometry, semantics, and dynamics – remains challenging, especially when 3D ground truth is unavailable. On the other hand, videos provide a rich supervisory signal: multiple views of the same scene over time for free. Hence, the goal of our work is to turn casual videos into structured, object-centric 3D scene representations.

Classical multi-view geometry algorithms achieve accurate 3D reconstruction under mostly static conditions through structure-from-motion and multi-view stereo, but these assumptions limit scale and generality [Hartley and A. Zisserman 2004; Schönberger and Frahm 2016]. In contrast, contemporary neural-network based

approaches learn semantics from large 2D datasets yet often lack explicit geometric consistency [Dosovitskiy et al. 2020; Vaswani et al. 2017]. A central objective of this work is to couple these strengths – geometric reliability with semantic breadth – so that understanding remains consistent across viewpoints and scenes.

Neural field representations provide a flexible substrate for this coupling: implicit, continuous models of radiance, occupancy, and features enable learning from images while representing 3D structure [Mildenhall et al. 2020]. To support object-centric understanding at scale, these fields can be informed by reliable 2D priors and vision-language supervision, bringing open-vocabulary semantics into 3D [Radford et al. 2021; Kerr et al. 2023].

Furthermore, real-world operation requires persistence through occlusion and motion. Video supervision naturally provides multi-view and temporal cues that align with deformable fields and long-range correspondence, helping maintain coherence over time [Park et al. 2021; Doersch et al. 2023].

In summary, this thesis is motivated by the goal of unifying geometry, semantics, and dynamics within a common framework for 3D scene understanding. It investigates how geometric priors, neural representations, and multimodal signals can convert everyday videos into structured, object-centric 3D descriptions that generalize across scenes and domains. We review the relevant literature in Section 1.2.

1.2 Key Ideas and Literature Review

1.2.1 From Classical multi-view Geometry to Learning based 3D Perception

Classical computer vision methods have established how to recover 3D structure from images using epipolar geometry, structure-from-motion (SfM), and multi-view stereo (MVS) [Hartley and A. Zisserman 2004; Schönberger and Frahm 2016; Schönberger et al. 2016; Yao et al. 2018]. These pipelines deliver accurate reconstructions, but many of these methods assume the camera calibration is fixed across a video and largely static scenes. Machine learning broadened the scope by learning to infer geometry and semantics directly from data, with deep neural

networks excelling at recognition while often lacking explicit geometric consistency [Dosovitskiy et al. 2020; Vaswani et al. 2017]. Bridging these paradigms, recent work leverages weak multi-view cues-differentiable warping, cross-view matching, and epipolar constraints to supervise models without dense 3D labels or poses [Tulsiani et al. 2017; Tulsiani et al. 2018; Rhodin et al. 2018; Facil et al. 2019; J. Sun et al. 2021; Sarlin et al. 2020; Y. He et al. 2020; Rockwell et al. 2022; Revaud et al. 2019; DeTone et al. 2018; Yi et al. 2016].

Incorporating geometry in learning can be done explicitly or implicitly. Explicit approaches inject geometric structure or camera parameters into the network (e.g. epipolar transformers and epipolar plane encodings) [Y. He et al. 2020; Yifan et al. 2022], while implicit approaches guide learning via multi-view consistency or correspondence losses without requiring geometry at test time [Tulsiani et al. 2018; Rhodin et al. 2018]. Robust estimation methods and improved correspondence (e.g. MAGSAC++, SuperPoint/LoFTR, SuperGlue) complement these trends by providing higher-quality supervision or pseudo-geometry [Barath et al. 2020; DeTone et al. 2018; J. Sun et al. 2021; Sarlin et al. 2020; Fischler and Bolles 1981].

1.2.2 Neural Fields and Implicit 3D Representations

Neural fields – continuous implicit functions parameterised by neural networks – have redefined 3D scene representation. Starting from radiance fields for view synthesis, variants now encode occupancy, signed distance, material, and semantics [Sitzmann et al. 2019; Mildenhall et al. 2020; L. Liu et al. 2020; Müller et al. 2022; A. Yu et al. 2021; Garbin et al. 2021; Reiser et al. 2021; Barron et al. 2021; Kerbl et al. 2023]. While highly expressive, many methods are scene-specific and assume known camera poses and static content. This has motivated research on efficiency, scalability, and compositional structure, including compact encodings and alternative primitives, as well as semantic fields that connect perception and reconstruction.

Extensions address unconstrained capture and dynamics, improving robustness to pose noise and scene changes [Martin-Brualla et al. 2021]. Efficiency-oriented designs (e.g. tensor decompositions and multiresolution structures) reduce memory and compute while retaining quality [A. Chen et al. 2022; A. Yu et al. 2021; Garbin et al. 2021; Reiser et al. 2021]. Alternative explicit primitives such as 3D Gaussian

Splatting further push real-time rendering and interactive applications, broadening downstream use in editing, 4D reconstruction, and generation [Kerbl et al. 2023].

1.2.3 Object-Centric, Multimodal, and Dynamic 3D Understanding

Real environments are composed of parts and objects arranged into scenes. Object-centric learning aims to recover this structure automatically, from early slot-based decompositions in 2D to 3D-aware, instance-level fields [Burgess et al. 2019; Locatello et al. 2020; Greff et al. 2019; Niemeyer and Geiger 2020; Schwarz et al. 2020; Y. Liu et al. 2023; Zarzar et al. 2022]. Building on these ideas, semantic and panoptic neural fields combine per-view labels into coherent 3D segmentations [Zhi et al. 2021a; Vora et al. 2021; Kundu et al. 2022; Siddiqui et al. 2023a; Y. Liu et al. 2023; Fu et al. 2022]. Large 2D segmentation models provide reliable priors that can be lifted into 3D [T.-Y. Lin et al. 2014; Gupta et al. 2019; Kirillov et al. 2023], and 3D feature/language fields enable open-vocabulary queries and semantic reasoning in 3D [Jia et al. 2021; Radford et al. 2021; Kerr et al. 2023; Peng et al. 2023; H. Chen et al. 2024; Engelmann et al. 2023; Jatavallabhula et al. 2023; Qin et al. 2023; Zuo et al. 2024].

Beyond static scenes, dynamics require modelling time-varying geometry and long-range correspondence. Deformable or dynamic fields capture non-rigid motion [Park et al. 2021; Pumarola et al. 2020; Park et al. 2020], while correspondence and tracking link observations over time using points or segments [Doersch et al. 2023; Karaev et al. 2023; H. K. Cheng et al. 2023; J. Wu et al. 2022]. Bridging these threads, 3D-aware tracking and fusion lift 2D evidence to geometric space for persistent identities under motion and occlusion [Plizzari et al. 2024].

Taken together, these threads point toward scalable, object-centric, and multimodal 3D understanding from casual video: combine geometric consistency from multi-view signals, the expressivity of neural fields, and the semantic breadth of foundation models, while extending from static scenes to dynamics. This perspective motivates the contributions outlined next.

1.3 Thesis Outline and Contributions

In this section, we outline the structure and contributions of the thesis. We provide an overview of each chapter and group the work into four themes: (i) Geometry-aware Learning, (ii) Object-centric 3D Learning, (iii) Hierarchical and Multimodal Understanding, and (iv) Dynamic Scene Understanding. Each chapter corresponds to a peer-reviewed publication and collectively advances the goal of learning structured, object-centric 3D representations directly from unconstrained videos.

Geometry-aware Learning

In Chapter 2, we investigate how to teach geometric reasoning to transformer-based vision models. While vision transformers trained on 2D images capture rich semantics, they lack the constraints required for consistent 3D reasoning. We introduce an epipolar-aware attention objective that guides transformer attention maps to follow geometric relations across views, enabling viewpoint-invariant object matching without requiring known camera poses at inference. Evaluated on a large-scale benchmark derived from CO3Dv2, the approach improves pose-invariant retrieval and generalises across unseen object categories. This demonstrates that geometric structure can be learned implicitly through weak, differentiable supervision.

Object-centric 3D Learning

In Chapter 3, we address the problem of reconstructing 3D object structure from 2D supervision. We propose a method for lifting 2D instance predictions from large segmentation models into 3D neural fields, producing coherent object reconstructions across multiple views. The approach fuses per-view masks through contrastive slow-fast alignment, achieving accurate 3D instance segmentation without any 3D labels. It performs strongly on both real and synthetic datasets, including a new "Messy Rooms" benchmark for multi-object scenes. This work demonstrates how 2D foundation models can provide scalable, geometry-consistent 3D understanding of complex environments.

Hierarchical and Multimodal Understanding

In Chapter 4, we extend static 3D representations to hierarchical and multimodal reasoning. We introduce Nested Neural Feature Fields (N2F2), a unified model that encodes parts, objects, and full scenes within a single continuous feature field. The model allocates subspaces to different semantic levels and aligns them with vision-language embeddings, enabling open-vocabulary querying and language-guided exploration of 3D environments. Compared to prior semantic neural fields, N2F2 achieves faster inference, higher accuracy, and supports flexible, hierarchical reasoning. This connects neural field representations with multimodal and language-grounded 3D understanding.

Dynamic Scene Understanding

In Chapter 5, we study dynamic scene understanding in egocentric videos. We develop a framework for 3D-aware instance segmentation and tracking that combines 2D segmentation, geometric lifting, and 3D association over time. Operating directly in 3D space enables consistent object identities through occlusions and viewpoint changes, while supporting amodal reconstruction of moving objects. This formulation unifies 3D perception with temporal reasoning, advancing toward persistent, object-centric world models that capture motion in realistic settings.

1.3.1 Publications

Chapters 2 to 5 each contain a paper that has been peer-reviewed and accepted at a conference. The papers are presented here in their original published form, the only difference being formatting.

For every publication, a corresponding authorship statement is attached. The papers included in the thesis are as follows:

- **Chapter 2: "A Light Touch Approach to Teaching Transformers Multi-view Geometry"** **Yash Bhalgat**, João F. Henriques, Andrew Zisserman. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.
- **Chapter 3: "Contrastive Lift: 3D Object Instance Segmentation by Slow-Fast Contrastive Fusion"** **Yash Bhalgat**, Iro Laina, João

F. Henriques, Andrew Zisserman, Andrea Vedaldi. In *Advances in Neural Information Processing Systems* (Spotlight), 2023.

- **Chapter 4: "N2F2: Hierarchical Scene Understanding with Nested Neural Feature Fields"** **Yash Bhalgat**, Iro Laina, João F. Henriques, Andrew Zisserman, Andrea Vedaldi. In *European Conference on Computer Vision*, 2024.
- **Chapter 5: "3D-Aware Instance Segmentation and Tracking in Egocentric Videos"** **Yash Bhalgat***, Vadim Tschernezki*, Iro Laina, João F. Henriques, Andrea Vedaldi, Andrew Zisserman. In *Asian Conference on Computer Vision*, 2024.

Publications not included. In addition, I have written these following works during the time of my DPhil but exclude them due to their lower relevance to the main topic of this thesis.

- **"Neural refinement for absolute pose regression with feature synthesis"** Shuai Chen, **Yash Bhalgat**, Xinghui Li, Jiawang Bian, Kejie Li, Zirui Wang, Victor Adrian Prisacariu. In *CVPR*, 2024.
- **"SiLVR: Scalable Lidar-Visual Reconstruction with Neural Radiance Fields for Robotic Inspection"** Yifu Tao, **Yash Bhalgat**, Lanke Frank Tarimo Fu, Matias Mattamala, Nived Chebrolu, Maurice Fallon. In *ICRA*, 2024.
- **"Reflecting Reality: Enabling Diffusion Models to Produce Faithful Mirror Reflections"** Ankit Dhiman, Manan Shah, Rishubh Parihar, **Yash Bhalgat**, Lokesh R Boregowda, R Venkatesh Babu. In *3DV*, 2025.
- **"GS-CPR: Efficient Camera Pose Refinement via 3D Gaussian Splatting"** Changkun Liu, Shuai Chen, **Yash Bhalgat**, Siyan Hu, Ming Cheng, Zirui Wang, Victor Adrian Prisacariu, Tristan Braud. In *ICLR*, 2025.
- **"Jamais Vu: Exposing the Generalization Gap in Supervised Semantic Correspondence"** Octave Mariotti, Zhipeng Du, **Yash Bhalgat**, Oisin Mac Aodha, Hakan Bilen. In *NeurIPS*, 2025.

* Joint first authorship contribution

Chapter 2

A Light Touch Approach to Teaching Transformers Multi-view Geometry

The paper was published at the Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

A Light Touch Approach to Teaching Transformers Multi-view Geometry

Yash Bhalgat João F. Henriques Andrew Zisserman

Visual Geometry Group

University of Oxford

{yashsb,joao,az}@robots.ox.ac.uk

May 4, 2026

Abstract

Transformers are powerful visual learners, in large part due to their conspicuous lack of manually-specified priors. This flexibility can be problematic in tasks that involve multiple-view geometry, due to the near-infinite possible variations in 3D shapes and viewpoints (requiring flexibility), and the precise nature of projective geometry (obeying rigid laws). To resolve this conundrum, we propose a "light touch" approach, guiding visual Transformers to learn multiple-view geometry but allowing them to break free when needed. We achieve this by using epipolar lines to guide the Transformer's cross-attention maps during training, penalizing attention values outside the epipolar lines and encouraging higher attention along these lines since they contain geometrically plausible matches. Unlike previous methods, our proposal does not require any camera pose information at test-time. We focus on pose-invariant object instance retrieval, where standard Transformer networks struggle, due to the large differences in viewpoint between query and retrieved images. Experimentally, our method outperforms state-of-the-art approaches at object retrieval, without needing pose information at test-time.

2.1 Introduction

Recent advances in computer vision have been characterized by using increasingly generic models fitted with large amounts of data, with attention-based models (e.g. Transformers) at one extreme [Dosovitskiy et al. 2020; Carion et al. 2020; Fedus et al. 2021; Ze Liu et al. 2021; Oquab et al. 2023; Jaegle et al. 2021]. There are many such recent examples, where shedding priors in favour of learning from more data has proven to be a successful strategy, from image classification [Dosovitskiy et al. 2020; Yuan et al. 2021; Han et al. 2021; Oquab et al. 2023; Ali et al. 2021], action recognition [Neimark et al. 2021; Girdhar et al. 2019; Plizzari et al. 2021; H. Fan et al. 2021; Bertasius et al. 2021], to text-image matching [Radford et al. 2021; Jia et al. 2021; W. Su et al. 2020; Lu et al. 2019] and 3D recognition [Zhao et al. 2021; K. Lin et al. 2021]. One area where this strategy has proven more difficult to apply is solving tasks that involve reasoning about multiple-view geometry, such as object retrieval – i.e. finding all instances of an object in a database given a single query image. This has applications in image search [Nie et al. 2007; Van Leuken et al. 2009; Jing and Baluja 2008; Krapac et al. 2010; W. Zhou et al. 2010], including identifying landmarks from images [Noh et al. 2017; Weyand et al. 2020; Radenović et al. 2018], recognizing artworks in images [Ufer et al. 2021], retrieving relevant product images in e-commerce databases [Oh Song et al. 2016; L. Cheng et al. 2020] or retrieving specific objects from a scene [Arandjelović and Andrew Zisserman 2011; Johnson et al. 2015; Rabinovich et al. 2007; J. Ma et al. 2020].

The main challenges in object retrieval include overcoming variations in viewpoint and scale. The difficulty in viewpoint-invariant object retrieval can be partially explained by the fact that it requires disambiguating similar objects by small differences in their unique details, which can have a smaller impact on an image than a large variation in viewpoint. For this reason, several works have emphasized geometric priors in deep networks that deal with multiple-view geometry [Facil et al. 2019; Yifan et al. 2022]. It is natural to ask whether these priors are too restrictive, and harm a network's ability to model the data when it deviates from the geometric assumptions. As a step in this direction, we explore how to "guide" attention-based networks with soft guardrails that encourage them to respect multi-view geometry, without constraining them with any rigid mechanism to do so.

Figure 2.1: Top-4 retrieved images with (1) global retrieval (left column), (2) Reranking Transformer (RRT) [Tan et al. 2021] (middle), and (3) RRT trained with our proposed Epipolar Loss (right column). Correct retrievals are **green**, incorrect ones are **red**. The Epipolar Loss imbues RRT with an implicit geometric understanding, allowing it to match images from extremely diverse viewpoints.

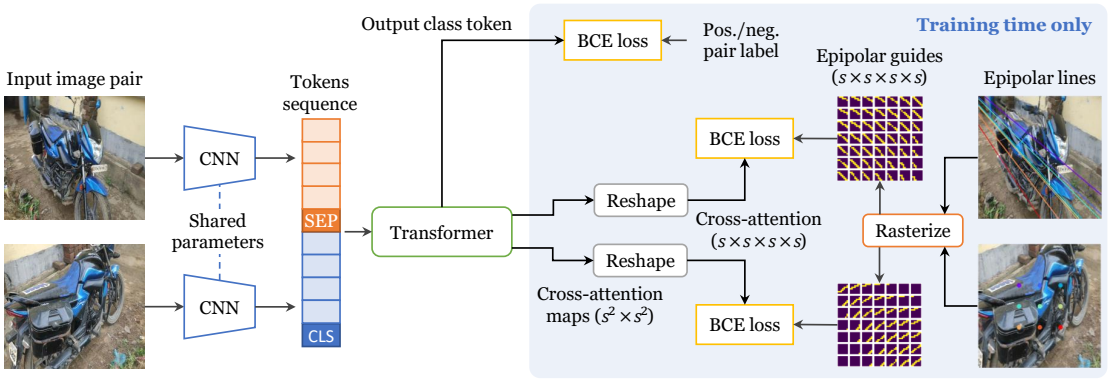


Figure 2.2: Overview of the proposed method. Features from two candidate images are extracted with a Convolutional Neural Network, and concatenated into a sequence of tokens for a Transformer. They are separated by a learned ⟨SEP⟩ token and end with a ⟨CLS⟩ token. The model is trained with a Binary Cross Entropy (BCE) loss to predict whether the two images match. During training, epipolar lines relating the two views (obtained with ground truth camera information) are rasterized into 4D tensors. These "epipolar guides" denote matches that are geometrically plausible given the viewpoints, and are used to train the Transformer's cross-attention maps using BCE losses.

In this work, we focus on post-retrieval reranking methods, wherein an initial ranking is obtained using global (image-level) representations and then local (region- or patch-level) representations are used to *rerank* the top-ranked images either with the classic Geometric Verification [Philbin et al. 2007], or by directly predicting similarity scores of image pairs using a trained deep network [Tan et al. 2021; Hausler et al. 2021]. Reranking can be easily combined with any other retrieval method while significantly boosting the precision of the underlying retrieval algorithm. Recently, PatchNetVLAD [Hausler et al. 2021], DELG [B. Cao et al. 2020], and Reranking Transformers [Tan et al. 2021] have shown that learned reranking can achieve state-of-the-art performance on object retrieval. We show that the performance of such reranking methods can be further improved by *implicitly* inducing geometric knowledge, specifically the epipolar relations between two images arising from relative pose, into the underlying image similarity computation.

This raises the question of whether multiple view relations should be incorporated into the two view architecture *explicitly* rather than *implicitly*. In the explicit case, the epipolar relations between the two images are supplied as inputs. For example, this is the approach taken in the Epipolar Transformers architecture [Y. He et al. 2020] where candidate correspondences are explicitly sampled along the epipolar line, and in [Yifan et al. 2022] where pixels are tagged with their epipolar planes using a Perceiver IO architecture [Jaegle et al. 2021]. The disadvantage

of the explicit approach is that epipolar geometry must be supplied at inference time, requiring a separate process for its computation, and being problematic when images are not of the same object (as the epipolar geometry is then not defined). In contrast, in the implicit approach the epipolar geometry is only required at training time and is applied as a loss to encourage the model to learn to (implicitly) take advantage of epipolar constraints when determining a match.

We bring the following three contributions in this work: First, we propose a simple but effective *Epipolar Loss* to induce epipolar constraints into the cross-attention layer(s) of transformer-based reranking models. We only need the relative pose (or epipolar geometry) information during training to provide the epipolar constraint. Once trained, the reranking model develops an implicit understanding of the relative geometry between any given image pair and can effectively match images containing an object instance from very diverse viewpoints *without* any additional input. Second, we set up an object retrieval benchmark on top of the CO3Dv2 [Reizenstein et al. 2021] dataset which contains ground-truth camera poses and provide a comprehensive evaluation of the proposed method, including a comparison between implicit and explicit incorporation of epipolar constraints. The benchmark configuration is detailed in Sec. 2.4. Third, we evaluate on the Stanford Online Products [Oh Song et al. 2016] dataset using both zero-shot and fine-tuning, outperforming previous methods on this standard object instance retrieval benchmark.

2.2 Related Work

Computing epipolar geometry. Estimating epipolar geometry given an image pair is a fairly broad problem, well-studied in multi-view geometry and computer vision [Hartley and A. Zisserman 2004]. Classic techniques involve predicting interest points and their descriptors [Mikolajczyk et al. 2005; Lowe 1999; Arandjelović and Andrew Zisserman 2012; Rosten and Drummond 2006; Rublee et al. 2011] in the images and finding point correspondences to estimate the relative geometry [Longuet-Higgins 1981; Nistér 2004]. Several learning based methods have been proposed to provide improved interest point detection and features, e.g. R2D2 [Revaud et al. 2019] SuperPoint [DeTone et al. 2018], LIFT [Yi et al. 2016] and

MagicPoint [DeTone et al. 2017]. These features along with learning based local matching methods [Wiles et al. 2021; J. Sun et al. 2021; Sarlin et al. 2020] and robust optimization methods [Barath et al. 2020; Brachmann et al. 2017; Fischler and Bolles 1981] form a powerful toolbox for relative geometry estimation. We use a combination of LoFTR [J. Sun et al. 2021] and MAGSAC++ [Barath et al. 2020] to generate pseudo-geometry information in one of our compared methods.

Incorporating epipolar geometry in Deep Learning. Recently, many works have proposed incorporating geometric priors into deep networks to deal with problems requiring multi-view understanding, such as 3D pose estimation [Y. He et al. 2020; F. Yu et al. 2021; Rhodin et al. 2018], 3D reconstruction [Yifan et al. 2022; Tulsiani et al. 2017] or depth estimation [Prasad et al. 2018]. Most of these approaches incorporate the epipolar geometry explicitly, e.g. Epipolar Transformers [Y. He et al. 2020] compute 3D-aware features for a point by aggregating features sampled on the corresponding epipolar line, which are shown to improve multi-view 3D human-pose estimation. [Yifan et al. 2022], another explicit method, proposed a few ways of featurizing multi-view geometry by encoding camera parameters or epipolar plane parameters and using them to provide geometric priors at the input-level. The epipolar plane encoding is also studied in this paper in the context of reranking transformers. Works such as [Rhodin et al. 2018; Tulsiani et al. 2018] propose implicitly incorporating geometric priors using multi-view consistency. Our work also falls in the implicit category, where we use epipolar constraints as a loss function applied to cross-attention maps to induce geometric understanding.

Image representations for retrieval. Traditionally, hand-crafted descriptors such as SIFT [Lowe 1999], RootSIFT [Arandjelović and Andrew Zisserman 2012] and BoVW [Philbin et al. 2007] were widely used for object retrieval. However, learned image-level (global) and region-level (local) representations [Arandjelovic et al. 2016; Babenko et al. 2014; Gordo et al. 2016; DeTone et al. 2018; Yi et al. 2016] have shown to surpass the performance of engineered features on large-scale datasets. Local learned representations can also be simply extracted as feature volumes from convolution neural networks or transformer backbones. Global representations are obtained by a combination of (1) downsampling/pooling operations inside a deep network, (2) learned clustering-based pooling operations [Arand-

jelovic et al. 2016] and/or specialized pooling operations such as R-MAC [Tolias et al. 2015]. Hybrid approaches that combine global and local features have also recently been proposed [Hausler et al. 2021; B. Cao et al. 2020].

Post-retrieval reranking. Early reranking methods, such as [Philbin et al. 2007; H. Jégou et al. 2008], used Geometric Verification (GV) with local features to compute geometric consistency between the query and reference images. This improved the precision of the top-ranked retrievals. Query Expansion (QE) was used to improve the recall. Popular QE variants such as average-QE and α-QE compute an updated query descriptor from the global descriptors of the top retrieved images [Ondrej Chum et al. 2007; Ondřej Chum et al. 2011; Tolias and Hervé Jégou 2014] and use it to retrieve a new set of top-ranked images. GV can be combined with many deep learning based retrieval methods used today, e.g. [B. Cao et al. 2020] uses RANSAC based GV on local features from its backbone model. Since RANSAC-based GV can be prohibitively slow for practical applications, [Hausler et al. 2021] proposes a rapid spatial scoring technique as an efficient alternative. Recently, transformer based methods [El-Nouby et al. 2021; Tan et al. 2021] have been introduced for retrieval and reranking. Our work builds on top of Reranking Transformers [Tan et al. 2021].

Retrieval with 3D information. Recently, methods using 3D data [Uy and Lee 2018; Zhe Liu et al. 2019], structural cues [Oertel et al. 2020] or view synthesis [Taira et al. 2018] have been proposed. Our goal in this work is to build image representations that capture 3D priors and can be used to retrieve images with large variations in pose or scale.

2.3 Method

We describe two variants of our method that *implicitly* or *explicitly* encourage Transformers to use geometric constraints in their predictions. Our work is built on top of Reranking Transformers (RRT) [Tan et al. 2021], a state-of-the-art approach for object retrieval with reranking. The explicit version, inspired by recent work [Yifan et al. 2022], serves both as a baseline and as a contrast to our proposed implicit approach. We first provide a brief review of RRTs for the reader (Sec. 2.3.1) and then describe our proposed Epipolar Loss (Sec. 2.3.3), as well as

Epipolar Positional Encodings (Sec. 2.3.4). The implementation details are given in Sec. 2.5.2.

2.3.1 Review of Reranking Transformers

Post-retrieval reranking is a popular technique used to boost the precision of object retrieval methods, wherein an initial ranking is obtained using global (image-level) descriptors and then local (region-level) descriptors along with the global ones are used to *rerank* the top-ranked images. In [Tan et al. 2021], each image ($\mathcal{I}$) is processed through a ResNet-50 [K. He et al. 2016] model to extract local features from the last convolution layer, with size $s \times s \times c$ ($s = 7$, $c = 2048$). Each of the s^2 local feature vectors is linearly projected from size c to a smaller size $m = 128$. Let these be denoted by $\mathbf{x}^l \in \mathcal{R}^{s^2 \times m}$ and their 2D positions in the feature volume by $\mathbf{p}_i \in \mathcal{R}^2$. The global features, computed as the mean of the local features, are used for initial ranking. Then, a light-weight transformer model (4 self-attention heads, 6 layers) is used to rerank these top predictions. With $\mathcal{I}$ as the query and $\bar{\mathcal{I}}$ as a reference image from the top predictions, as well as class $\langle$CLS$\rangle$ and separator $\langle$SEP$\rangle$ tokens (consisting of learnable embeddings), the input to the transformer model is constructed as the concatenation of tokens:

$$X(\mathcal{I}, \bar{\mathcal{I}}) = [\langle \text{CLS} \rangle, f(\mathbf{x}^l_1), \dots f(\mathbf{x}^l_{s^2}), \\ \langle \text{SEP} \rangle, \bar{f}(\bar{\mathbf{x}}^l_1), \dots \bar{f}(\bar{\mathbf{x}}^l_{s^2})]$$

where $f(\mathbf{x}^l_i) = \mathbf{x}^l_i + \psi(\mathbf{p}_i) + \beta$, $\bar{f}(\bar{\mathbf{x}}^l_i) = \bar{\mathbf{x}}^l_i + \psi(\bar{\mathbf{p}}_i) + \bar{\beta}$, $\psi(.)$ is the frequency position encoding [Vaswani et al. 2017] and $\beta, \bar{\beta}$ are learnable embeddings that differentiate descriptors of $\mathcal{I}, \bar{\mathcal{I}}$. Sec. 2.5.2 provides training details for the reranking model.

2.3.2 Review of Epipolar Geometry

Epipolar geometry limits the possible image correspondences for projections of an observed 3D point from different viewpoints. A central concept is the epipolar line. Consider a 2D point $\mathbf{x}$ in one image. It may correspond to an infinity of 3D points – one for each possible depth – which lie on a 3D line that extends from the camera center and passes through $\mathbf{x}$ in the image plane. This 3D line, when projected into a *second* image captured from another viewpoint, is an epipolar line of $\mathbf{x}$. This

mapping from a point in one image to its epipolar line in another image can be seen in Fig. 2.2 (right). Epipolar geometry can be used to effectively constrain matches across viewpoints: starting from a point in one image, it can only match points in another image that lie along its epipolar line. Epipolar geometry can be computed directly from two images, either from their relative pose or from correspondences, without requiring any information about depth or 3D geometry of the observed scene. Mathematically it is represented by a 3×3 fundamental matrix. For a more detailed exposition, please refer to [Hartley and A. Zisserman 2004].

2.3.3 Epipolar Loss

Given the feature volumes $\mathbf{x}^l, \bar{\mathbf{x}}^l \in \mathcal{R}^{s^2 \times m}$ as input tokens (in addition to $\langle$CLS$\rangle$, $\langle$SEP$\rangle$), let $\mathbf{y}_{L-1}, \bar{\mathbf{y}}_{L-1}$ denote the corresponding inputs to the last transformer layer in the RRT model. The *raw* cross-attention between these outputs can be computed as $A^{12} = Q\overline{K}^T$ and $A^{21} = \overline{Q}K^T$, where W_Q, W_K are query and key projection matrices and $Q = W_Q\mathbf{y}_{L-1}$, $K = W_K\mathbf{y}_{L-1}, \overline{Q} = W_Q\bar{\mathbf{y}}_{L-1}, \overline{K} = W_K\bar{\mathbf{y}}_{L-1}$.

Next, given the epipolar geometry between the input images, for every location $i \in \{1, \ldots, s^2\}$ in $\mathbf{x}^l$, we can find the set of locations $\bar{e}_i$ in $\bar{\mathbf{x}}^l$ that lie on the corresponding epipolar line. Similarly, for each location $i \in \{1, \ldots, s^2\}$ in $\bar{\mathbf{x}}^l$, we can find the corresponding set of locations e_i in $\mathbf{x}^l$. We want to encourage the network, for a given position in the first volume, to only *attend* to corresponding epipolar positions in the other volume. This is done by penalizing attention values that have high values outside the epipolar lines, and encouraging the attention along epipolar lines to be high. This is achieved by using a Binary Cross Entropy (BCE) loss on the *raw* cross-attention maps $\{A^{12}, A^{21}\}$:

$$
\begin{aligned}
L^{12}(i,j) &= \text{BCE}(\sigma(A^{12}(i,j)), \mathbb{1}(i,j)) \\
L^{21}(i,j) &= \text{BCE}(\sigma(A^{21}(i,j)), \mathbb{1}(i,j)) \\
L_{EPI} &= \sum_{i=1}^{s^2} \sum_{j=1}^{s^2} L^{12}(i,j) + L^{21}(i,j)
\end{aligned} \tag{2.1}
$$

where σ is a sigmoid function, and $\mathbb{1}(i,j)$ is a special indicator function that is 1 when location j in the other feature map lies on the epipolar line corresponding to location i in the current map. The training process is illustrated in Fig. 2.2.

Max-Epipolar Loss. In the Epipolar Loss proposed above, every point on the corresponding epipolar line is encouraged to have high attention even if it is not the *actual* matching point in 3D. We also propose a variant called *Max-Epipolar Loss*, wherein we select only the point on the epipolar line with the maximum predicted cross-attention value and encourage the attention for that point to be high.

$$L_{MaxEPI} = L_{zero} + L_{max} \tag{2.2}$$

where

$$L_{max} = \sum_{i} \text{BCE}\left(\max_{j \in e_i} \sigma(A(i,j)), 1\right)$$
$$L_{zero} = \sum_{\forall i,j, \mathbb{1}(i,j)=0} \text{BCE}(\sigma(A(i,j)), 0)$$

where e_i is the set of locations in the other feature map that lie on the epipolar line corresponding to location i in the current map. L_{zero}, L_{max} are applied to both A^{12} and A^{21}.

Note that the epipolar loss is applied at training time, so epipolar geometry is required only during training. The epipolar geometry can be obtained from the relative pose between the images, or from the images directly. However, as will be shown in Sec. 2.5.6, once trained with our proposed L_{EPI}, the attention map extracted from the trained RRT model for a previously *unseen* pair of images shows patterns corresponding to the actual epipolar lines (*without* any input epipolar geometry information). This demonstrates that the model's predictions are epipolar-geometry-aware, and at test time this leads to improved reranking performance as erroneous point matches can be avoided.

2.3.4 Epipolar Positional Encoding

In contrast to Epipolar Loss where we implicitly induce awareness of epipolar line correspondence into the model, epipolar constraints can be encoded *explicitly* by annotating each pixel with an encoding that uniquely identifies its epipolar plane. The family of epipolar planes "rotates" about the line joining the two camera centers, hence it can be parameterized by a scalar angle of rotation. Inspired by [Yifan et al. 2022], we introduce a baseline wherein we encode the epipolar plane

angle for each token and add the encoding to the input tokens of the transformer. A random epipolar plane corresponding to a randomly chosen pixel location is used as reference to calculate the plane angle. We encode the angle with the frequency positional encoding [Vaswani et al. 2017; Mildenhall et al. 2020].

The drawback of the explicit method is that it requires the epipolar geometry (or relative pose) information during inference. In the scenario when this information is not available, we have to rely on other ways to obtain the epipolar geometry or relative pose which may not be entirely accurate and leads to loss in performance, as will be shown in Sec. 2.5.3. In fact, determining whether epipolar geometry can be established between two views is essentially replacing the job of the reranking transformer in determining if two images contain the same object.

2.4 CO3D-Retrieve benchmark

We now describe how we repurpose the CO3Dv2 [Reizenstein et al. 2021] dataset to create a large-scale object instance retrieval benchmark with multiple views of real objects. CO3Dv2 is a dataset of multi-view images of common object categories, consisting of 36,506 videos of object instances (one video per object instance), taken from distant viewpoints spanning all 360 degrees, and covering 51 common object categories. The dataset also contains the ground-truth camera poses for the video frames and foreground segmentation masks for the object in each image.

For the *CO3D-Retrieve* dataset, we extract 5 frames per video so that each frame is separated from the next by *approximately* 72° of rotation around the object. In total, CO3D-Retrieve contains 181,857 images of 36,506 object instances. We split the dataset into two halves for training and testing: the training dataset contains 91,106 images of 18,241 object instances, and the testing datasets contains 90,751 images from 18,265 object instances. The set of object instances seen during training and testing are disjoint and so have zero overlap with each other. For benchmarking object retrieval on CO3D-Retrieve, we evaluate with each image as the query, the other images from the same object as the query are treated as *positives*, and all the images not corresponding to the query object are treated as negatives. Fig. 2.3 shows example object images from the benchmark.

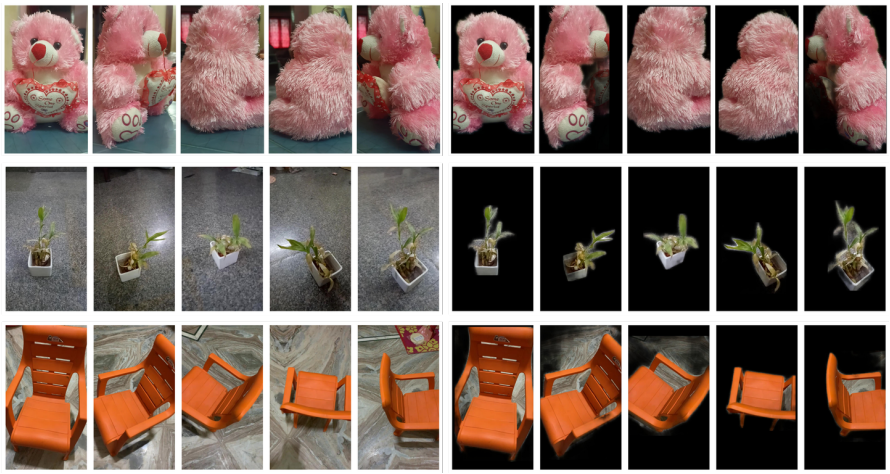


Figure 2.3: Example images for three object instances from the CO3D-Retrieve benchmark. The left half shows the full image, and the right half shows the masked counterparts obtained using the object masks in CO3D [Reizenstein et al. 2021]. The number of pixels in common between views of the same object decreases from the top row to the bottom (computed using the 3D point-clouds, also available from CO3D).

2.5 Experiments

In this section, we evaluate the epipolar-geometry aware Reranking Transformer on two datasets: our CO3D-Retrieve benchmark, and the Stanford Online Products (SOP) benchmark [Oh Song et al. 2016]. SOP is a popular benchmark for object retrieval containing 120,053 images of 22,634 object instances from 12 object categories. We use the standard train-test split used by all the baselines we compare with, where 59,551 images are used for training and 60,502 for testing. In Sec. 2.5.6, we provide a discussion on the merits of the implicit approach to incorporating epipolar constraints and explore its properties.

2.5.1 Baselines and metrics

Pretrained descriptors. Deep networks pretrained on large scale image datasets learn powerful image representations that can be used for retrieval. Evaluating such pretrained models without fine-tuning gives us a lower bound on the performance that a model trained on our dataset should achieve. We compare with VGG16 [Simonyan and Andrew Zisserman 2015] and ResNet50 (R50) [K. He et al. 2016] models pretrained on ImageNet [Deng et al. 2009], i.e. trained for classification, not retrieval. We also compare to a NetVLAD [Arandjelovic et al. 2016] model (i.e. VGG16 backbone + NetVLAD pooling) pretrained for retrieval on Pittsburgh250k [Torii et al. 2013].

Reranking Transformers (RRT). Reranking Tranformers (RRT) [Tan et al. 2021] is a state-of-the-art method that our works builds on. We compare with different versions of the RRT method:

1. *R50 (trained)*: this baseline performs global retrieval (no reranking) and does not use RRT, but works as a foundation for subsequent baselines. The model is trained using a batch-wise contrastive loss on CO3D-Retrieve or SOP [Oh Song et al. 2016], for the respective experiments.
2. *R50 (frozen) + RRT*: we start from a trained R50 (i.e. baseline (1)), freeze its weights and train a RRT on top of it for reranking.
3. *R50 (finetune) + RRT*: we start from a trained R50 (i.e. baseline (1)) and we finetune the R50 backbone along with the RRT.

RRT w/ Epipolar Positional Encoding. The R50 backbone along with RRT is trained with their respective "retrieval loss" functions, and the epipolar geometry is provided as input in the form of an Epipolar Positional Encoding (Sec. 2.3.4). We will discuss the results of this baseline in a separate Sec. 2.5.5.

Evaluation metrics. Given a query image and a retrieved image, they match if they contain the same object instance. We report two metrics to evaluate retrieval performance. First, **R@K** – for a given query, if a match is within the top K retrieved images, then the query is said to be retrieved. $R@K$ is the fraction of correctly retrieved queries for a given K. We report results for $K = 1$, 10 and 50. Second, we report **Mean Average Precision (mAP)** – the mean of the Average Precision [Manning et al. 2008] over all queries.

2.5.2 Implementation details

Extracting the epipolar geometry. For experiments with the CO3D-Retrieve benchmark, the available ground-truth pose information for each image is used to compute the epipolar geometry for a matching pair of images. During training, we use the "RandomCrop" image augmentation which shifts the principal point location. Hence, we adjust the Fundamental Matrix computation appropriately to

obtain the correct epipolar lines in the cropped images. When pose information is not available as ground-truth, such as in the SOP [Oh Song et al. 2016] dataset, we use an off-the-shelf method to compute the epipolar geometry. Specifically, we use a local image feature matching method, LoFTR [J. Sun et al. 2021], to extract high-quality semi-dense matches between the image pair. Then, we use a robust estimation method, MAGSAC++ [Barath et al. 2020] to extract the Fundamental Matrix. The epipolar geometry extracted with this method is not entirely accurate (especially for image pairs with extreme relative pose), but it provides us with a sufficient pseudo ground-truth epipolar geometry to train our models with Epipolar Loss. If the number of matches found ≤ 20 or number of inliers detected $\leq 0.2\times$ number of matches, we consider the extracted epipolar geometry unreliable and do not apply Epipolar Loss for that image pair during training. Computing epipolar geometry with this method takes $\approx$0.06 seconds per image pair on a 8-core CPU and NVIDIA P40 GPU.

Training details. We use a ResNet50 [K. He et al. 2016] for global retrieval, which is trained with a batchwise contrastive loss (batch size of 800). For an image pair$\{\mathcal{I}, \bar{\mathcal{I}}\}$ in the batch, a Binary Cross-Entropy loss is used to train the reranking model enforcing its output to be 1 if $\mathcal{I}$ and $\bar{\mathcal{I}}$ contain the same object and 0 otherwise.

In our proposed implicit method, the global retrieval model and the reranking model are trained with their respective retrieval-losses plus the Epipolar Loss. If $\mathcal{I}$ and $\bar{\mathcal{I}}$ represent the same object, then the epipolar geometry between the image pair (which is extracted as explained above) is used to compute the Epipolar Loss for training. If the image pair is not a match, then a valid epipolar geometry does not exist and we simply do not apply the Epipolar Loss for that image pair.

In the explicit method, we have to include the geometry in the input as Epipolar Positional Encodings (EPE), even when the input pair $\{\mathcal{I}, \bar{\mathcal{I}}\}$ is not a match. To handle the case when $\{\mathcal{I}, \bar{\mathcal{I}}\}$ is not a match during training and testing, we use a *random* rank-2 matrix as the Fundamental Matrix to compute the EPEs. When $\{\mathcal{I}, \bar{\mathcal{I}}\}$ is indeed a match, (a) during training, we use the ground-truth or the pseudo ground-truth (whichever is available) to compute the EPEs, (b) during testing, we do not rely on the ground-truth geometry information and always use the LoFTR/MAGSAC++ method (described above) to compute the EPEs.

Table 2.1: Evaluation on CO3D-Retrieve benchmark. Description of all compared methods in Sec. 2.5.1. Baselines shown above dashed line are pretrained models and below are trained on CO3D-Retrieve. **EPE**=Epipolar Positional Encoding

Method	Full images				With masked backgrounds			
	R@1	*R*@10	*R*@50	mAP	*R*@1	*R*@10	*R*@50	mAP
Pretrained Models								
VGG16 [Simonyan and Andrew Zisserman 2015]	66.21	85.18	91.66	22.51	63.56	81.11	89.24	16.73
R50 [K. He et al. 2016]	66.48	85.34	91.74	22.79	63.81	80.30	89.37	16.79
NetVLAD [Arandjelovic et al. 2016]	67.01	85.17	91.72	22.63	63.19	80.84	89.27	16.61
R50 (trained)	86.06	95.62	97.65	45.34	78.82	91.30	94.72	24.85
RRT [Tan et al. 2021] + R50 (frozen)	88.07	96.29	97.75	47.60	82.45	91.89	94.85	26.16
RRT + R50 (finetune) (**SOTA**)	89.20	96.85	97.89	48.81	83.28	92.13	95.05	27.33
RRT + R50 w/ EPE	88.53	96.41	97.83	47.99	82.79	91.96	94.99	26.58
RRT + R50 w/ L_{EPI} (**Ours**)	90.57	97.33	98.10	49.52	85.07	92.42	95.11	28.07
RRT + R50 w/ L_{MaxEPI} (**Ours**)	**90.69**	**97.38**	**98.10**	**49.60**	**85.17**	**92.46**	**95.14**	**28.21**

The hyperparameters we use for our experiments with SOP [Oh Song et al. 2016] are the same as [Tan et al. 2021], except that we use 40 epochs (instead of 100) when training with the Epipolar Loss with a constant learning rate of 10^{-4}. Hyperparameters used with CO3D-Retrieve are provided in Section 2.7.3. Our method is entirely implemented in PyTorch [Paszke et al. 2019].

2.5.3 Results on CO3D-Retrieve

We evaluate on CO3D-Retrieve in two settings: with and without masking the background in the object images. This is because the background also provides useful visual cues for image matching and it is essential to see how the methods perform without any such extra information. Figure 2.3 shows examples with and without masking the background.

Table 2.1 shows the detailed results. We observe that pretrained models (VGG16 and R50 on ImageNet, NetVLAD on Pittsburgh250k) achieve a reasonable performance without any finetuning. However, their performance is not competitive compared with baselines specialized for CO3D-Retrieve. It's interesting to note that a simple ResNet50-based global retrieval baseline trained with batch-wise contrastive loss (i.e. the "R50 (trained)" baseline) already achieves a high *R*@1 of 86.06% on the unmasked images. Our method, which uses the Epipolar Loss to induce multi-view geometric understanding in to the Reranking Transformer model, outperforms the state-of-the-art approach (RRT + R50 (finetune)) in both masked and unmasked settings. The margin with which the Epipolar Loss baseline outperforms "RRT + R50 (finetune)" is greater in the case of images with masked

Table 2.2: Evaluation on Stanford Online Products [Oh Song et al. 2016]. Baselines shown above the dashed line are pretrained models and below the dashed line are trained on SOP. More details in Sec. 2.5.1. Key: *=results obtained using checkpoints from [Tan et al. 2021]; **=results reported in [Tan et al. 2021]

Method	R@1	R@10	R@50	mAP
Pretrained Models				
VGG16 [Simonyan and Andrew Zisserman 2015]	55.75	70.86	79.65	11.93
R50 [K. He et al. 2016]	55.89	71.32	79.69	12.09
NetVLAD [Arandjelovic et al. 2016]	54.16	70.85	79.62	11.90
R50 (trained)*	80.74	91.87	95.54	32.90
RRT [Tan et al. 2021] + R50 (frozen)*	81.80	92.35	95.78	34.91
RRT + R50 (finetune)* (**SOTA**)	84.46	93.21	**96.04**	37.14
RRT + R50 w/ EPE	82.57	92.69	95.89	35.38
RRT + R50 w/ L_{EPI} (**Ours**)	**84.74**	**93.29**	**96.04**	**37.25**
RRT + R50 w/ L_{MaxEPI} (**Ours**)	84.53	93.27	**96.04**	37.19
*Other Metric Learning methods***				
Margin-based [Roth et al. 2020]	76.1	88.4	-	-
FastAP [Cakir et al. 2019]	73.8	88.0	-	-
XBM [X. Wang et al. 2020]	80.6	91.6	-	-
Cross-Entropy based [Boudiaf et al. 2020]	81.1	91.7	-	-

Table 2.3: Zero-shot evaluation on Stanford Online Products [Oh Song et al. 2016] with models trained on CO3D-Retrieve.

Method	R@1	R@10	R@50	mAP
RRT + R50 (frozen)	75.53	89.43	95.01	29.27
RRT + R50 (finetune)	76.32	90.16	95.21	30.19
RRT + R50 w/ L_{EPI} (**Ours**)	**76.78**	**90.27**	**95.29**	**30.25**

background, as this is a harder task. It can be seen that the Max-Epipolar variant of the Epipolar loss consistently gives a slight improvement.

2.5.4 Results on Stanford Online Products

The Stanford Online Products (SOP) dataset [Oh Song et al. 2016] does not contain ground-truth pose information for the object images. As detailed in Sec. 2.5.2, we obtain the pseudo ground-truth geometry information using LoFTR [J. Sun et al. 2021] for matching and MAGSAC++ [Barath et al. 2020] for robust estimation. We find that, even though these pseudo ground-truth poses are not entirely accurate, they are still useful for training with the Epipolar Loss. Table 2.2 shows a comprehensive comparison of our proposed methods with all the baselines. We also include deep metric learning methods [Roth et al. 2020; Cakir et al.

2019; X. Wang et al. 2020; Boudiaf et al. 2020] with reported numbers taken from [Tan et al. 2021] into our comparisons. Our proposed implicit method outperforms all the baselines including the state-of-the-art Reranking Transformers [Tan et al. 2021].

We also test zero-shot retrieval, by evaluating on SOP models that were trained on CO3D-Retrieve. The results are shown in Table 2.3, where we can observe that the differences between all methods are reduced, but our Epipolar Loss still confers a performance advantage.

2.5.5 Implicit vs Explicit methods

The transformer model trained with Epipolar Loss does not require pose or epipolar geometry information at test time. The explicit method, however, requires the fundamental matrix at the input (during both training and testing) to generate the Epipolar Positional Encodings (EPE). Tables 2.1 and 2.2 show that using EPE with Reranking Transformer adversely affects the performance, compared to not using the encodings. Although reasons for this decrease are unclear, one possibility is that the encodings leak information about whether two images match or not, because when they do not match, the input epipolar encodings are arbitrary. The network may learn to rely on this signal instead of image matching, in a case of "shortcut learning" [Geirhos et al. 2020]. This issue does not affect the implicit method as geometry information isn ot required at test time. When training the RRT, the Epipolar Loss is used with *only* those image pairs that contain matching images. Hence, during inference, the Transformer uses implicit geometry information only when it is *valid* (i.e. inherently for matching image pairs).

2.5.6 What does the implicit model learn?

After training with the Epipolar Loss (L_{EPI}) or the Max-Epipolar Loss (L_{MaxEPI}), we investigate if the attention maps of the learned model show some signs of geometric-awareness. To do this, we pick two matching images $\{\mathcal{I}, \bar{\mathcal{I}}\}$ from the *test* set, i.e. these images were not seen during training, and extract the cross-attention maps from the last layer of the transformer. Since we use a $7 \times 7 \times 128$ feature volume for each image (reshaped to 49×128 for the transformer), the cross-attention maps (from $\mathcal{I}$ to $\bar{\mathcal{I}}$, and $\bar{\mathcal{I}}$ to $\mathcal{I}$) are of size 49×49 which are then

reshaped back to $7 \times 7 \times 7 \times 7$. These $7 \times 7 \times 7 \times 7$ cross-attention map values indicate the attention between each *feature-pixel* of the first and second feature volume.

Fig. 2.5 shows predicted cross-attention maps alongside expected ground-truth maps. The latter are computed using ground-truth pose information. We observe that the attention maps obtained with L_{EPI} closely follow ground truth epipolar lines, despite this instance and associated geometry not being seen during training. Note that the attention maps obtained with L_{MaxEPI} are much sparser with peaks that lie on actual epipolar lines.

Do cross-attention maps learned with Max-Epipolar Loss correspond to true matching points? We overlay the cross-attention maps, extracted from the RRT [Tan et al. 2021] model trained with Max-Epipolar Loss (L_{MaxEPI}), on the original images to see if the location of highest attention coincides with the position of the actual matching point. Fig. 2.4 shows one such visual illustration. We can see that the peaks of attention maps *very loosely* coincide with the actual matching points on the corresponding ground-truth epipolar lines.

Additionally, we visualize the cross-attention maps for a pair of *mismatched* images, as shown in Fig. 2.6. These cross-attention maps are extracted from a RRT model trained with Epipolar Loss (L_{EPI}).

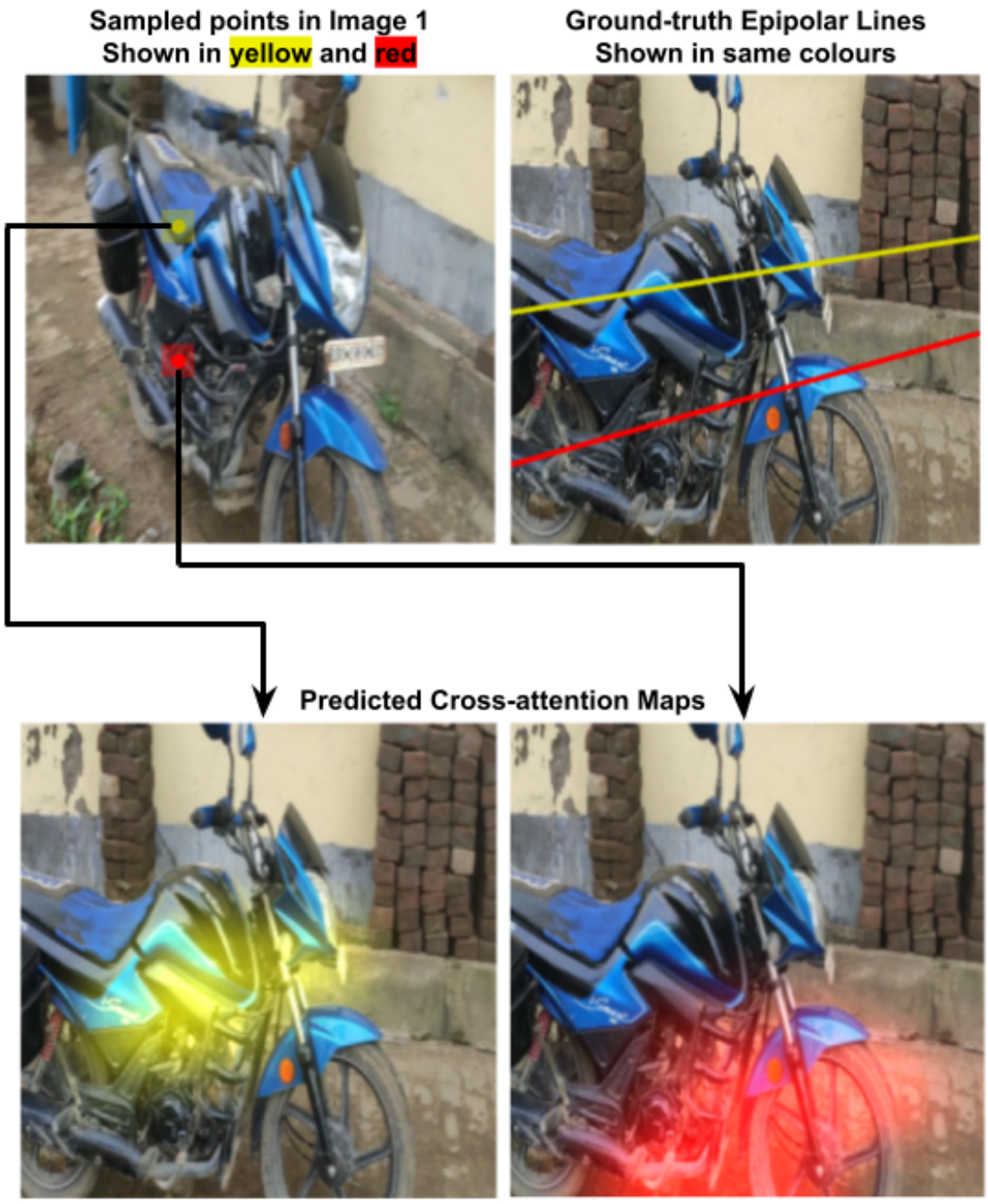


Figure 2.4: Cross-attention maps, extracted from the RRT model trained with Max-Epipolar Loss, overlaid on the image. Note: the attention maps are bilinearly upsampled to the size of the image.

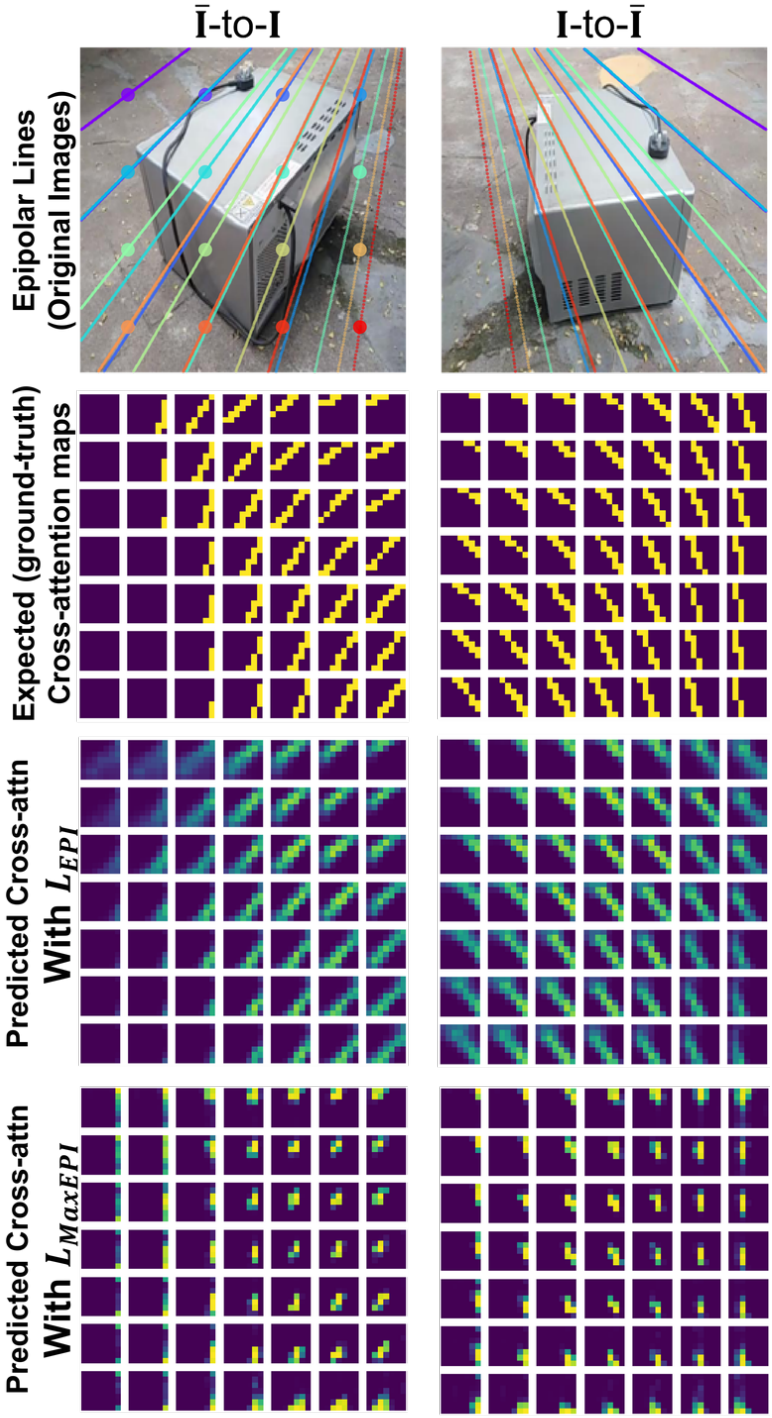


Figure 2.5: Visualization for a test image pair (never seen in training). **Top row**: Points shown in image $\mathcal{I}$ have correspondences on the epipolar lines of that color in image $\bar{\mathcal{I}}$. **Second row**: Expected 7×7×7×7 cross-attention maps shown as a 7×7 grid with a 7×7 patch at each grid location computed from the ground truth epipolar geometry. In the $\mathcal{I}$-to-$\bar{\mathcal{I}}$ grid (right column), a patch at grid location (i, j) shows the epipolar line in $\bar{\mathcal{I}}$ corresponding to the pixel (i, j) in the 7×7 feature space of $\mathcal{I}$. **Third row**: Predicted cross-attention maps from transformer trained with L_{EPI}. Notice how closely they match ground-truth maps, even though these are test images and don't have access to the ground truth epipolar geometry. **Bottom row**: Predicted cross-attention maps from transformer trained with Max-Epipolar Loss, L_{MaxEPI}. These are sparser and have peaks close to actual epipolar lines.

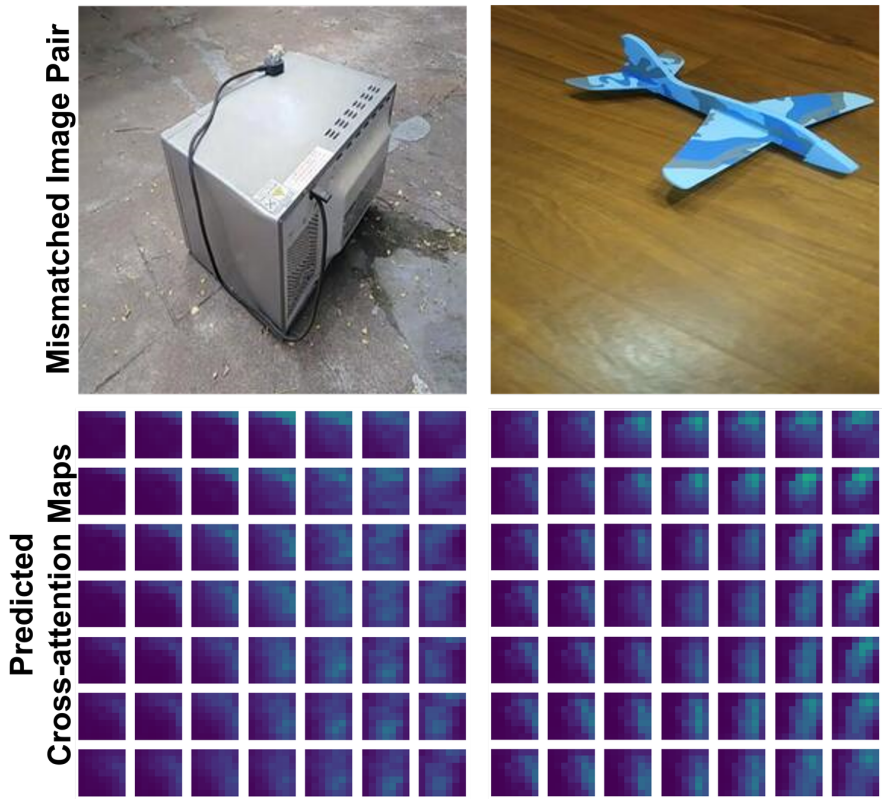


Figure 2.6: Predicted attention maps for a non-matching test image pair. A valid epipolar geometry does not exist for this pair, hence the model predicts *diffuse* attention maps.

2.6 Conclusion

In this work, we aimed to teach multi-view geometry to Transformer networks, and proposed a method to do so implicitly via epipolar guides. The advantages of this implicit approach over explicitly passing in geometric information to a network are two-fold: (i) ground-truth epipolar geometry (relative pose) between views is only needed at training time, not at inference; (ii) implicit losses are readily applied to existing architectures, so there is no need to design specialized architectures. We demonstrated improved performance over the state-of-the-art in object retrieval, by reranking with our method. More generally, this approach of implicitly incorporating knowledge into Transformers by a suitable loss can be employed in other scenarios. Examples include learning other geometric relations, such as a trifocal relationship over three views, as well as physical laws such as Newton's laws of motion.

Acknowledgements. We are grateful for funding from EPSRC AIMS CDT EP/S024050/1, AWS, the Royal Academy of Engineering (RF\201819\18\163), EPSRC Programme Grant VisualAI EP/T028572/1, and a Royal Society Research Professorship RP\R1\191132. We thank the authors of [Tan et al. 2021; J. Sun et al. 2021; Barath et al. 2020] for open-sourcing their code. We also thank an anonymous reviewer for useful suggestions on the Max-Epipolar Loss.

2.7 Appendix

2.7.1 Qualitative examples

In Figures 2.8-2.18, we provide a few qualitative results on the CO3D-Retrieve benchmark (Figures 2.8-2.12) and the Stanford Online Products [Oh Song et al. 2016] dataset (Figures 2.13-2.18). We visually compare the top-5 retrievals obtained with the Global Retrieval (R50) model, Reranking Transformer [Tan et al. 2021] model and a reranking model trained with our Epipolar Loss. We also accompany each example with its corresponding Precision-Recall curve, which provides a more detailed perspective on the retrieval performance.

In the CO3D-Retrieve benchmark, the *maximum* number of reference images per query is 4. So, for all the examples shown, the Precision-Recall curve for our

method saturates at Precision = 1.0 since the top-4 retrievals are correct.

2.7.2 Visualization of Attention Maps

For the sake of clarity, we describe, with an example, how the cross-attention map predicted by our Transformer model (trained with Epipolar Loss) contains information about the true epipolar geometry between the input image pair. Figure 2.19 shows such an example, where we select two points in a 7×7 grid (because the feature map extracted by our backbone is spatially 7×7) of the first image and show the actual (ground-truth) as well as the predicted epipolar lines in the other image.

2.7.3 Implementation details CO3D-Retrieve

For experiments with the CO3D-Retrieve benchmark, the global-retrieval-only model (R50 (trained) as described in Sec. 5.1) is trained for 50 epochs with the Adam optimizer and a learning rate that starts at 0.0001 and decays exponentially by a factor of 10 every 20 epochs. The Reranking Transformer head is trained on top of this trained global model, by either freezing or finetuning the global model, with or without the Epipolar Loss. When training without the Epipolar Loss, the model is trained using a SGD optimizer with an initial learning rate of 5×10^{-5} decayed exponentially by a factor of 10 over 40 epochs. When training with the Epipolar Loss, the above procedure is followed without Epipolar Loss for 20 epochs and the model is trained for an additional 20 epochs with Epipolar Loss with a learning rate of 10^{-6}. As mentioned in Section 2.5.2, the hyperparameters we use for our experiments with SOP [Oh Song et al. 2016] are the same as [Tan et al. 2021], except that we use 40 epochs (instead of 100 in [Tan et al. 2021]) when training with the Epipolar Loss with a constant learning rate of 10^{-4}.

2.7.4 mAP analysis with same category retrieval

We empirically observe that a majority of high-ranked (i.e. top-5) false positives are images from the same category. We conduct an experiment where we compute the mAP while ranking only the images from the same category as the query image and ignoring out-of-class images. As shown in Table 2.4, we get a higher mAP when retrieval is only performed on the same category images. This means there

Dataset Name	Same Category Retrieval	Full Dataset Retrieval
CO3D-Retrieve	52.03	49.52
SOP [Oh Song et al. 2016]	38.61	37.25

Table 2.4: Comparison of mAP computed while ranking only the images from the same category as the query image.

are confusing images from *outside* the categories, albeit a small fraction compared to intra-class.

Further per-category analysis with our method reveals that the *top 5* categories with the highest proportions of intra-class false positives (in descending order) are `banana, suitcase, laptop, keyboard, umbrella` in CO3D-Retrieve dataset and `fan, cabinet, mug, coffee_maker, kettle` in SOP [Oh Song et al. 2016].

2.7.5 Breakdown of $R@K$ based on fraction of overlapping pixels

The proposed CO3D-Retrieve benchmark includes large variations in viewing angle among images of the same instance. We conduct an analysis to understand how our proposed method performs under a range of pose variations. To do this, we first compute an "Overlap Score (OS)" for each instance using ground-truth point-clouds available in CO3D-Retrieve. Large pose-variations lead to a low OS. We divide the query-set into 10 bins uniformly between OS= 0.2 to 0.8 and compute the $R@1$ for each bin. These limits of OS are chosen because there are very few ($< 1\%$) instance with an OS beyond the $[0.2, 0.8]$ range. Figure 2.7 shows $R@1$ for our proposed method and RRT [Tan et al. 2021] with respect to the OS. We can see that the R@1 of our method drops by 5.5% from highest to lowest OS, while RRT [Tan et al. 2021] drops by 10.6%. In conclusion, our Epipolar Loss is useful in extreme viewpoint changes.

2.7.6 High-resolution results

In our experiments throughout the paper, the local features tensor obtained from Resnet50 backbone has a spatial resolution of 7×7. We train another transformer model with L_{EPI} on input images of size 448×448 so that we obtain 14×14

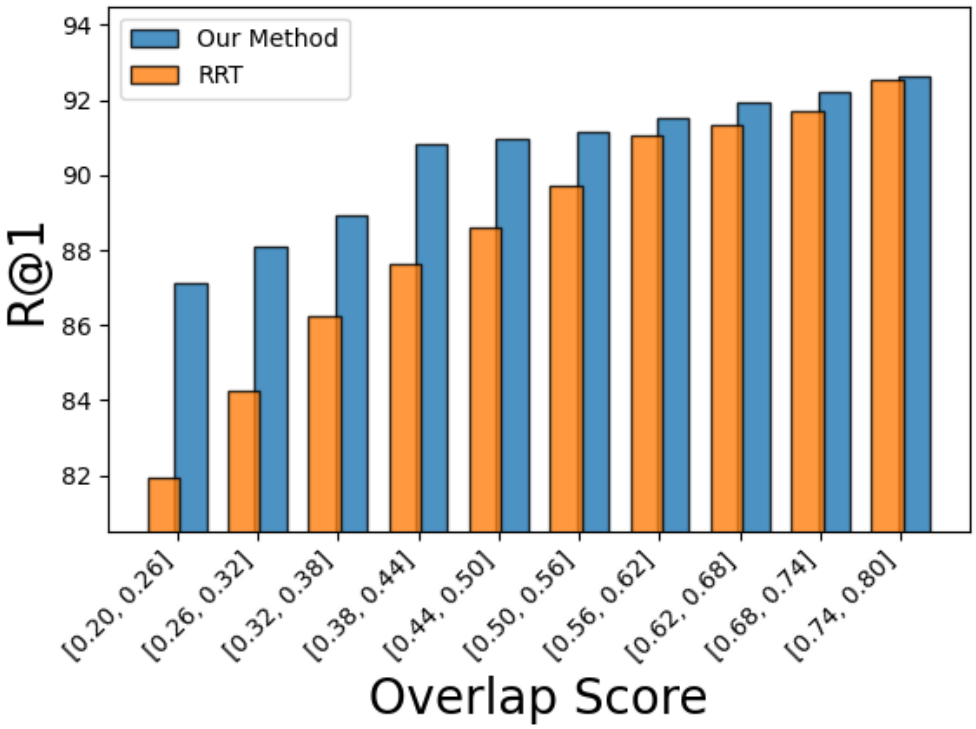


Figure 2.7: Breakdown of R@1 according to the Overlap Score of instances in the CO3D-Retrieve benchmark. The query set is divided into bins based on OS; these bins are shown on the x-axis.

Model	Local features resolution	R@1	R@10	R@50
Original	7×7	90.57	97.33	98.10
High-res	14×14	90.71	97.42	98.15

Table 2.5: Comparison with transformer model trained on 448×448 images. Both models are trained with L_{EPI}. "Original" corresponds to the result in Table 2.1.

local features. By doing this, we can obtain higher resolution ($14 \times 14 \times 14 \times 14$) cross-attention maps, as shown in Figure 2.20. Table 2.5 shows the performance achieved by the high resolution transformer model.

2.7.7 Failure Cases

It is important to look at the cases where our proposed method fails to retrieve good matches and analyze them for further improvement. Figures 2.21 and 2.22 show a few such examples for CO3D-Retrieve and SOP [Oh Song et al. 2016] respectively. We see that a common failure scenario for our method is when the query image is a close-up of the object (Fig. 2.21 (c,d) and Fig. 2.22 (c,d)) or repetitive patterns in objects such as keyboards (Fig. 2.21 (d)). A critical future direction for our work is to make the model robust to these scenarios.

2.7.8 Quality of Epipolar Geometry with LoFTR + MAGSAC++ method

During training, when the ground-truth epipolar geometry is not available, we use a *pseudo*-geometry predicted using a pretrained LoFTR [J. Sun et al. 2021] model for matching and MAGSAC++ [Barath et al. 2020] for robust optimization. The quality of the predicted epipolar geometry depends on the quality and number of matches obtained by the LoFTR model. In Figure 2.23, we show two examples demonstrating the success and failure cases of this method.

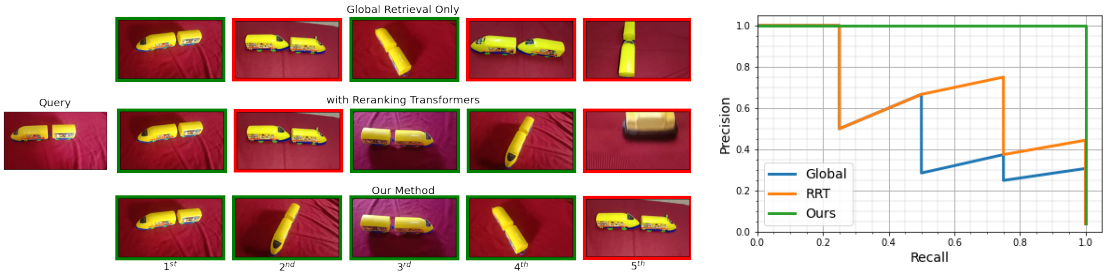


Figure 2.8: CO3D-Retrieve: Example 1.

Figure 2.9: CO3D-Retrieve: Example 2.

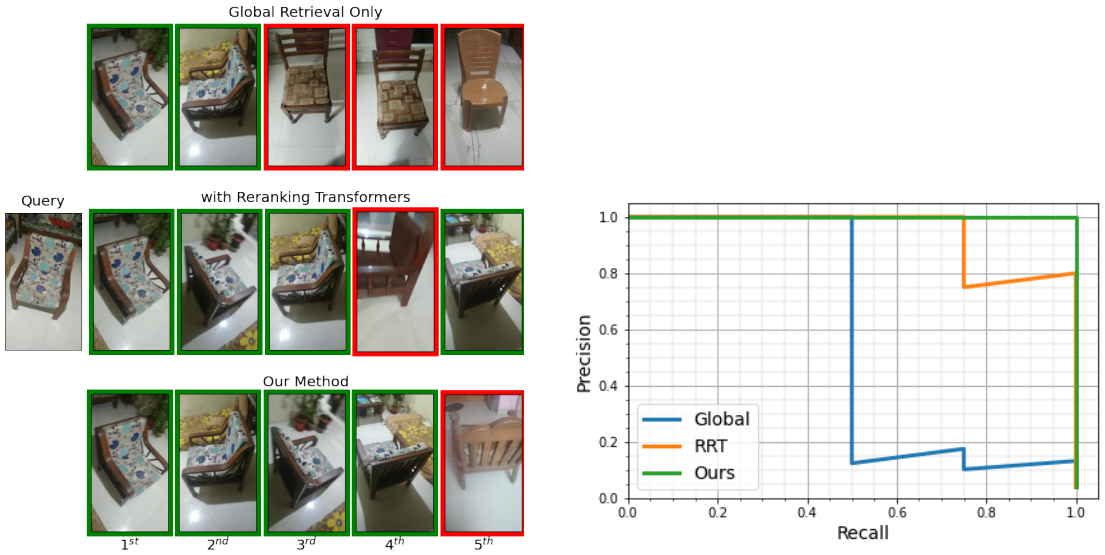


Figure 2.10: CO3D-Retrieve: Example 3.

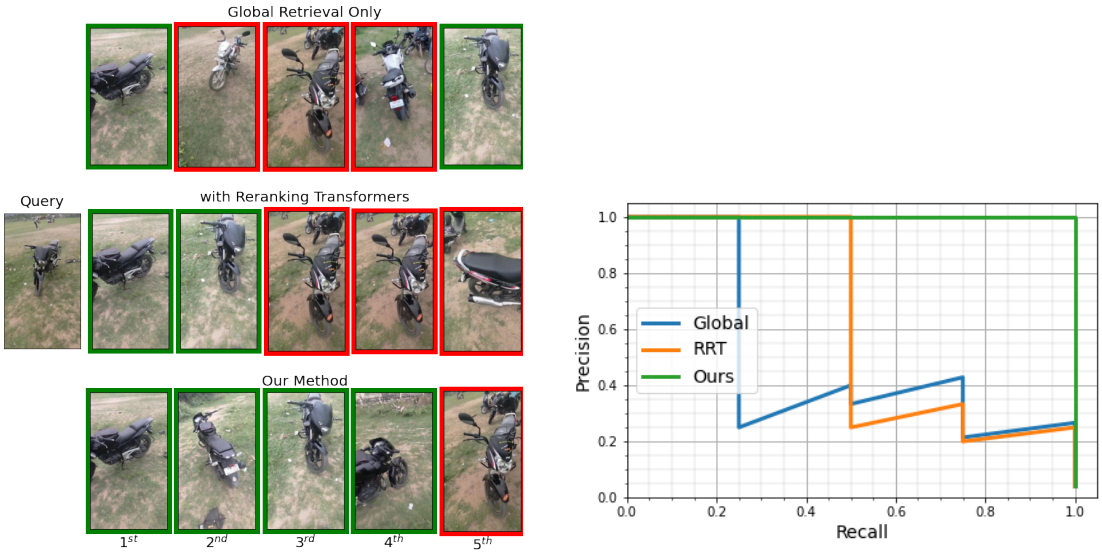


Figure 2.11: CO3D-Retrieve: Example 4.

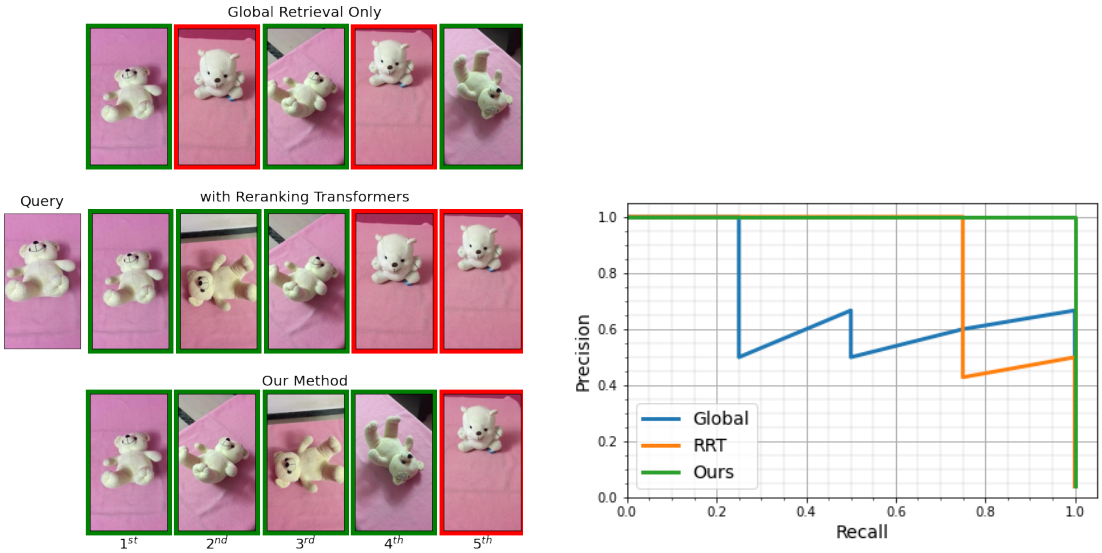


Figure 2.12: CO3D-Retrieve: Example 5.

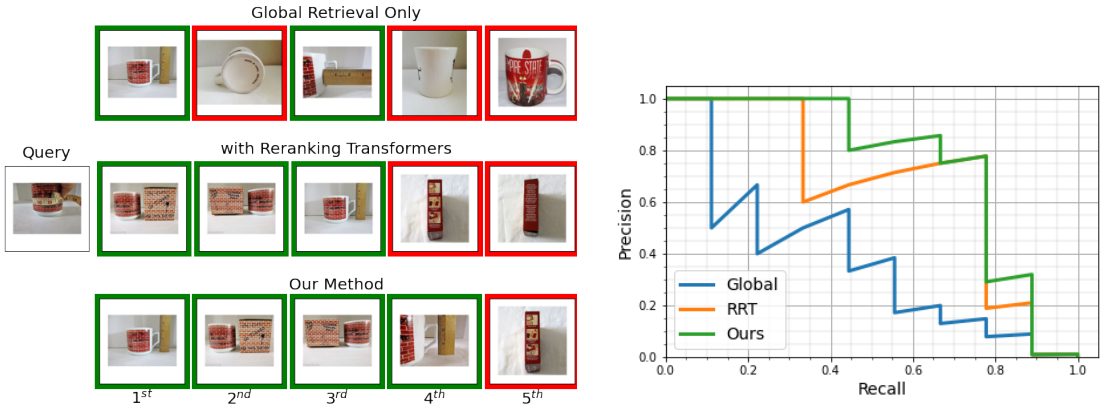


Figure 2.13: SOP [Oh Song et al. 2016] dataset. Example 1.

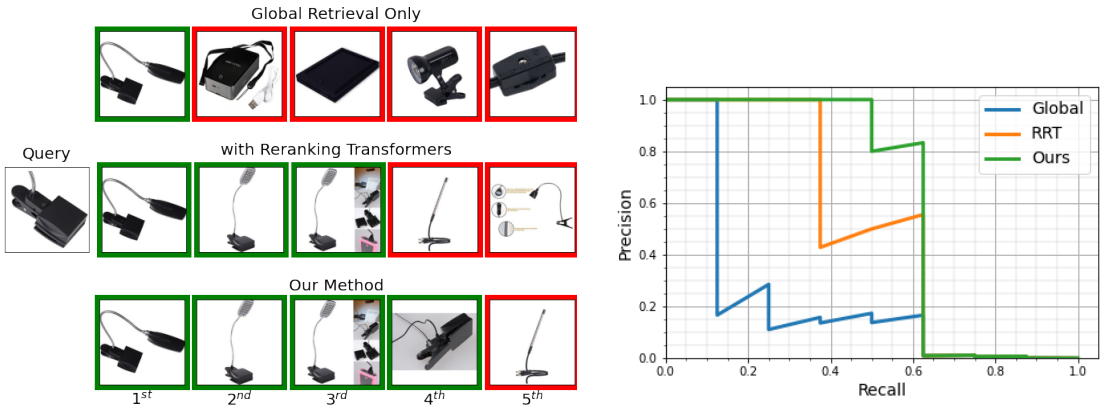


Figure 2.14: SOP [Oh Song et al. 2016] dataset. Example 2.

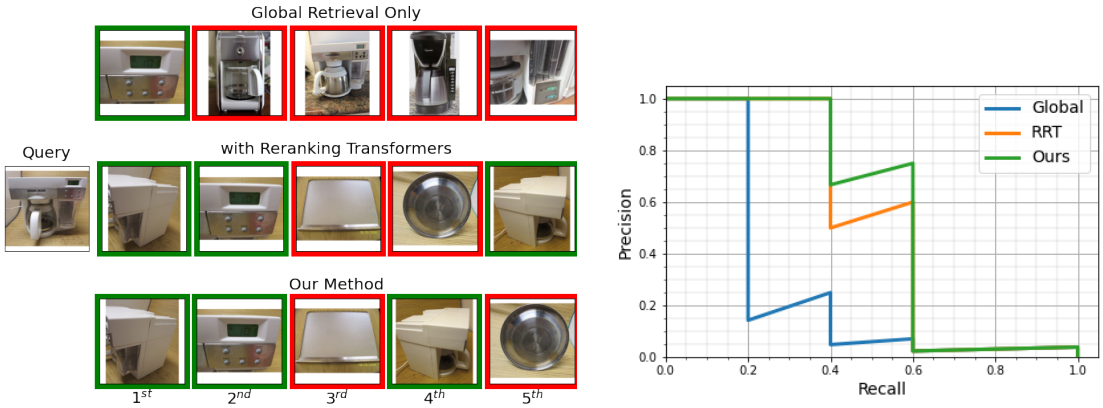


Figure 2.15: SOP [Oh Song et al. 2016] dataset. Example 3.

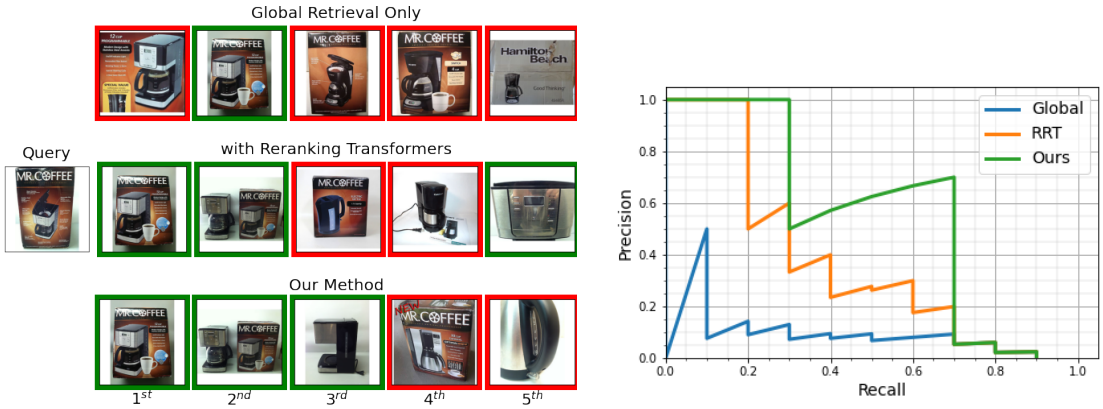


Figure 2.16: SOP [Oh Song et al. 2016] dataset. Example 4.

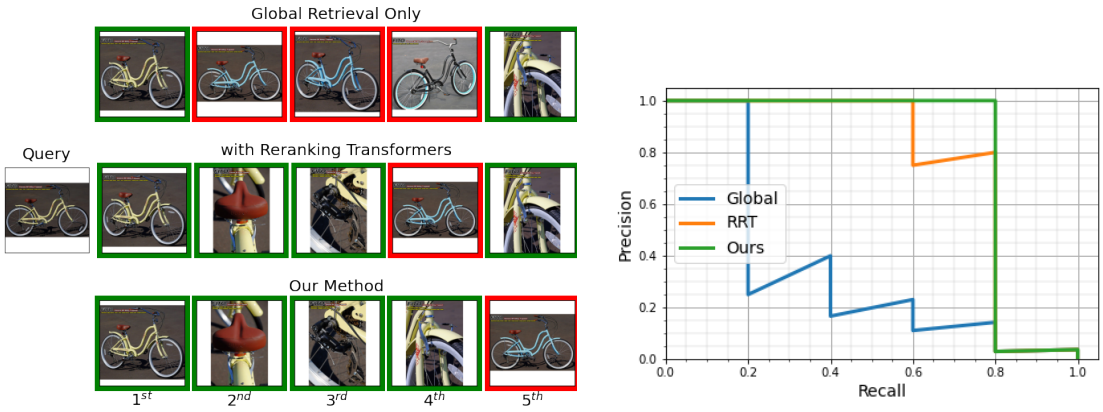


Figure 2.17: SOP [Oh Song et al. 2016] dataset. Example 5.

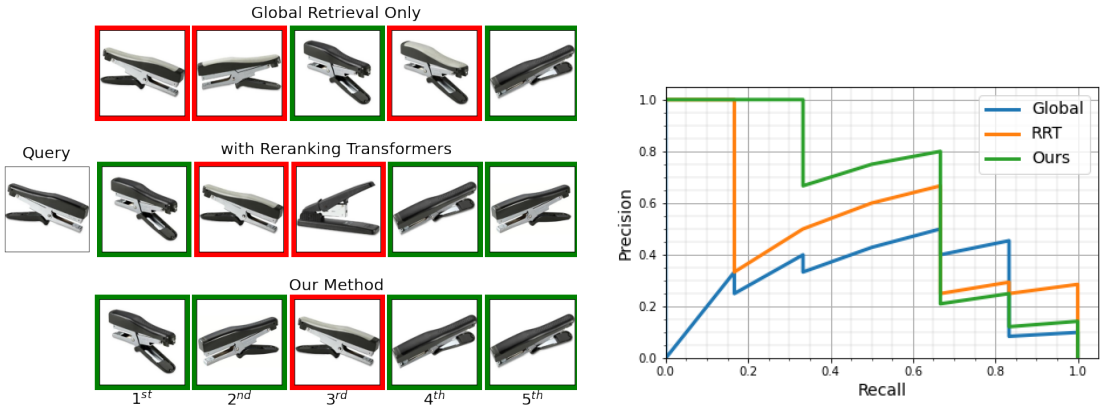
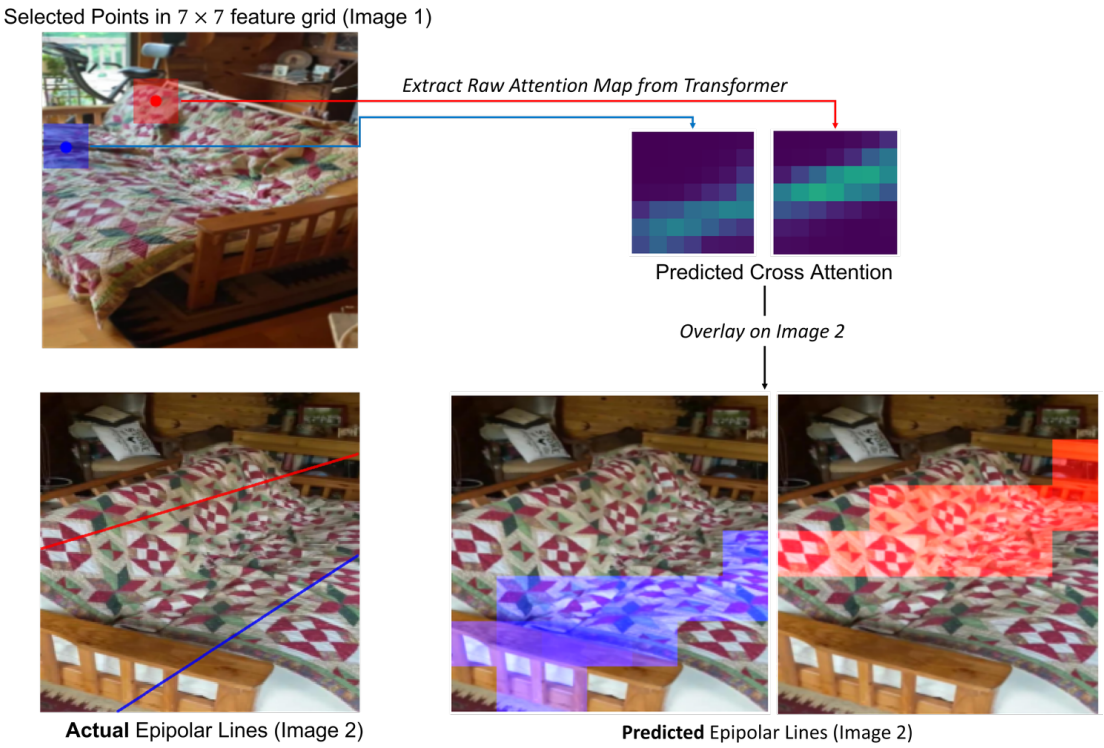
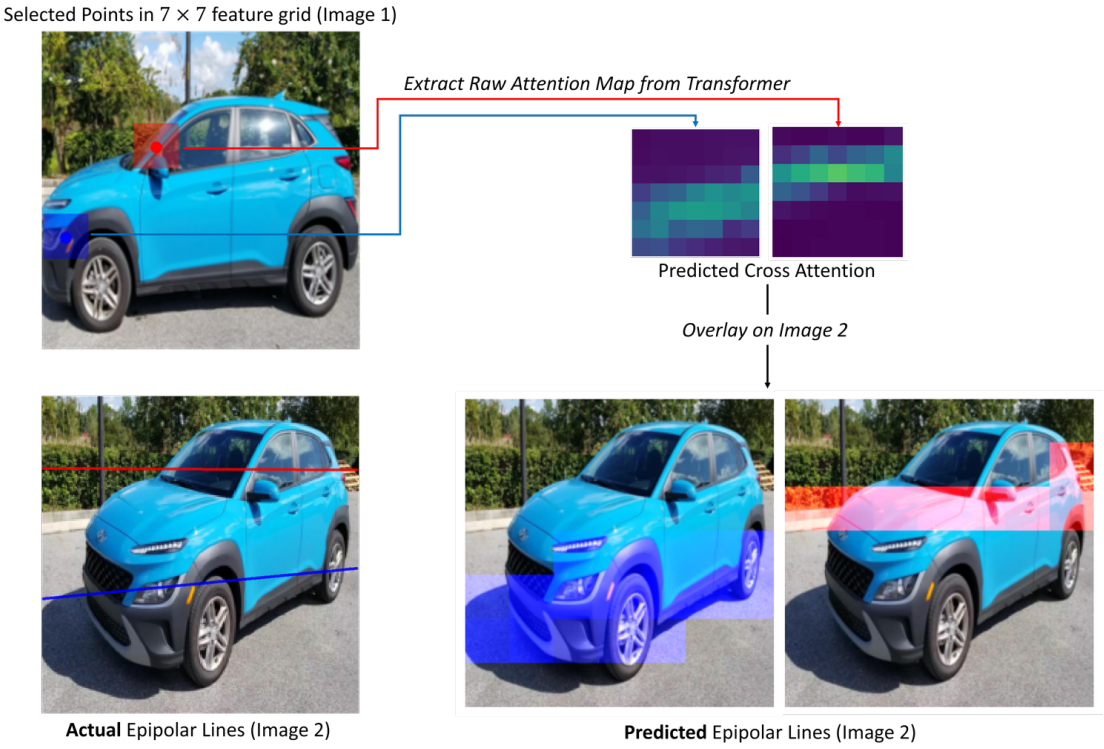


Figure 2.18: SOP [Oh Song et al. 2016] dataset. Example 6.

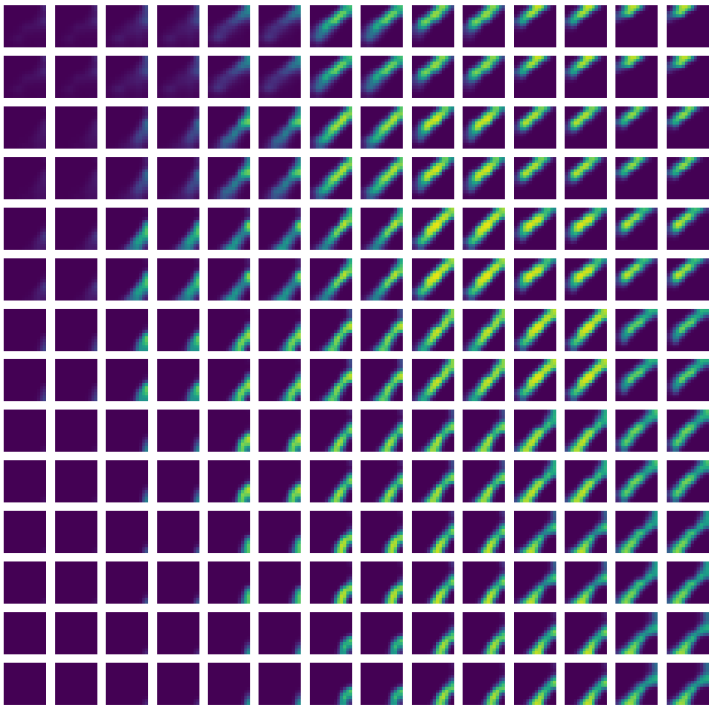


(a)

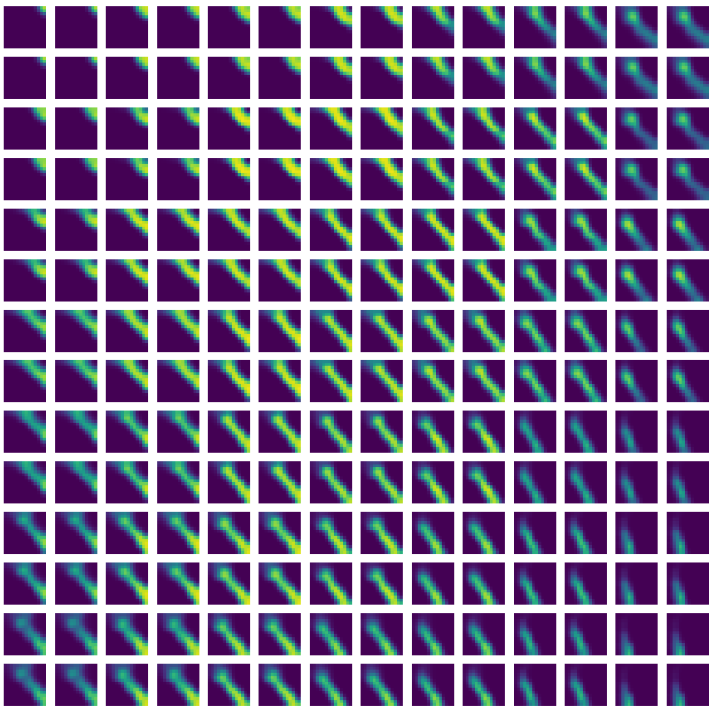


(b)

Figure 2.19: Examples showing how the cross-attention map predicted by our transformer model (trained with Epipolar Loss) contains information about the true epipolar geometry. Red and Blue colours are used to show the two selected points and their corresponding actual vs predicted epipolar lines.



(a) Cross-attention from $\bar{\mathbf{I}} \rightarrow \mathbf{I}$



(b) Cross-attention from $\mathbf{I} \rightarrow \bar{\mathbf{I}}$

Figure 2.20: Cross-attention maps extracted from the transformer model trained with 448×448 input images. Due to the higher input resolution, the cross-attention maps are obtained at a higher resolution of $14 \times 14 \times 14 \times 14$.

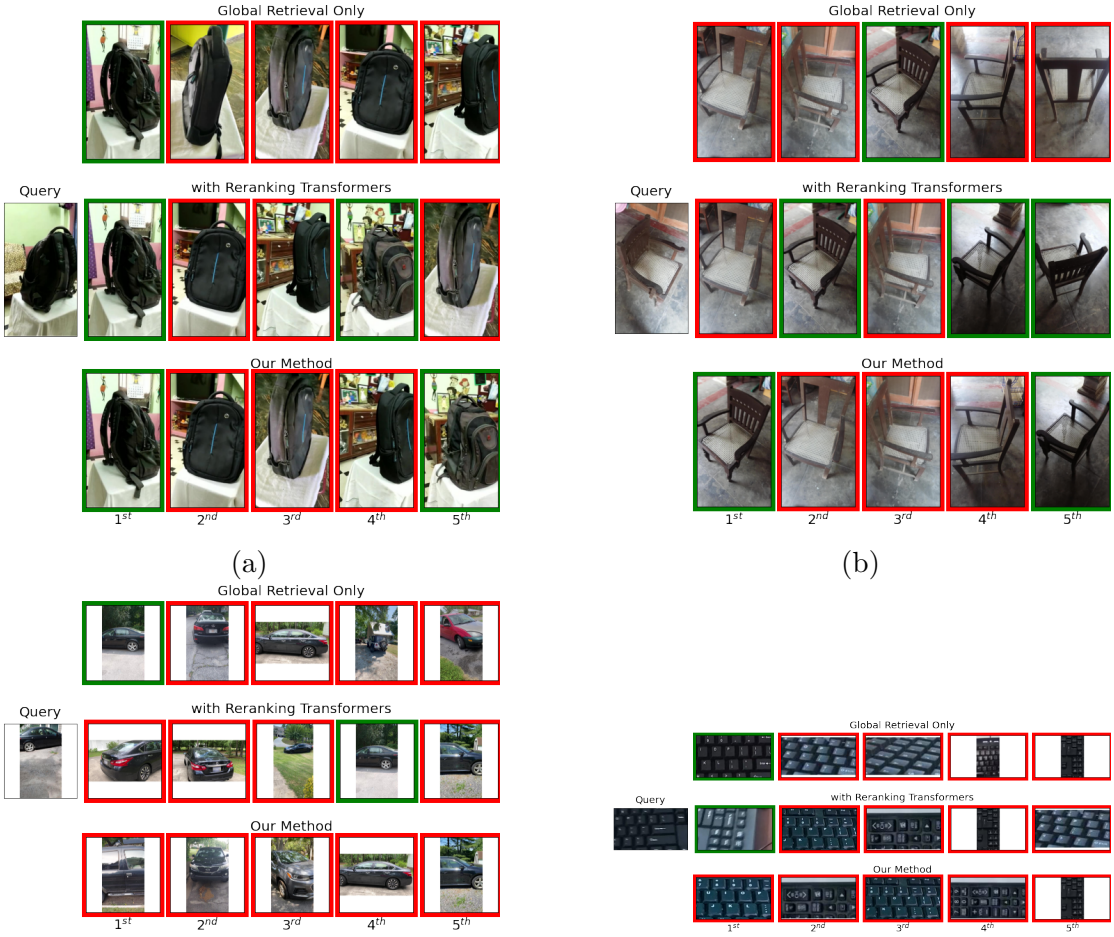


Figure 2.21: Failure cases from the CO3D-Retrieve dataset.

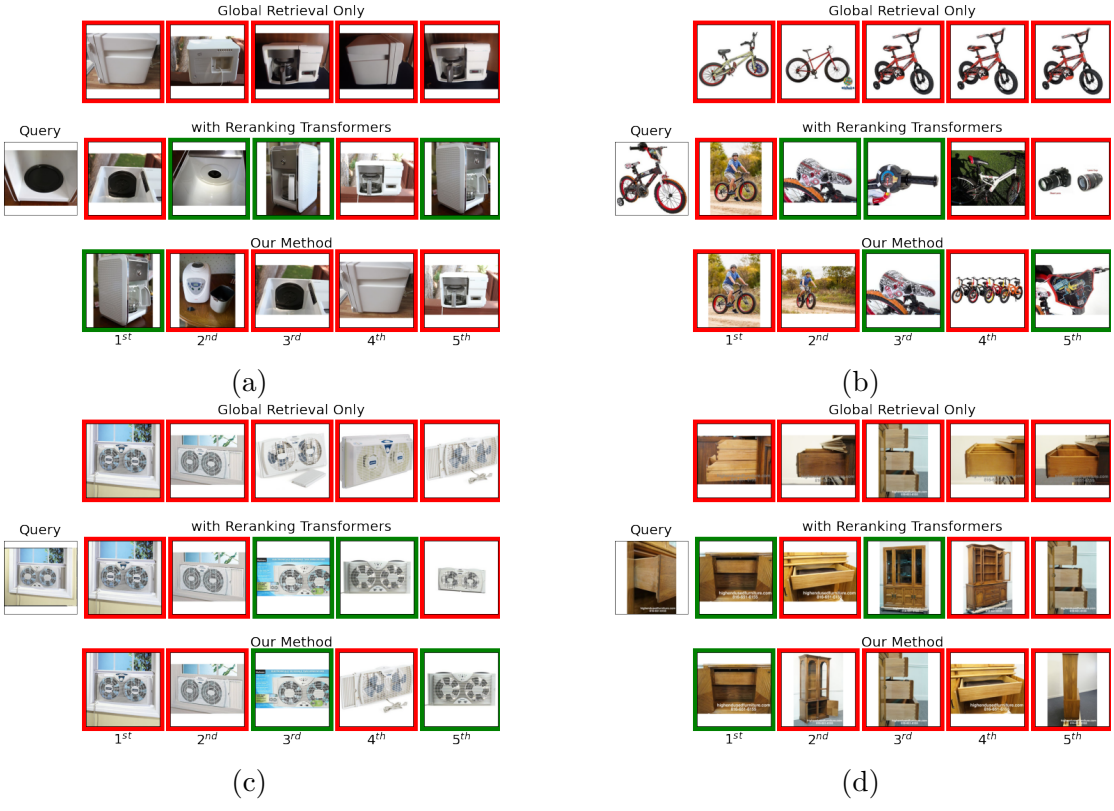
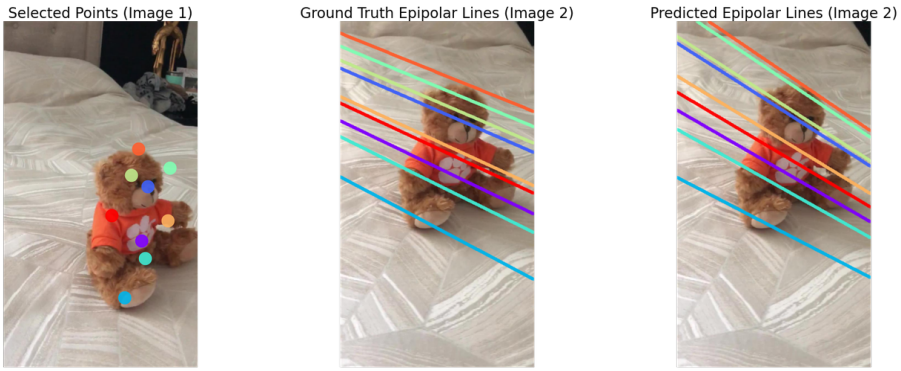
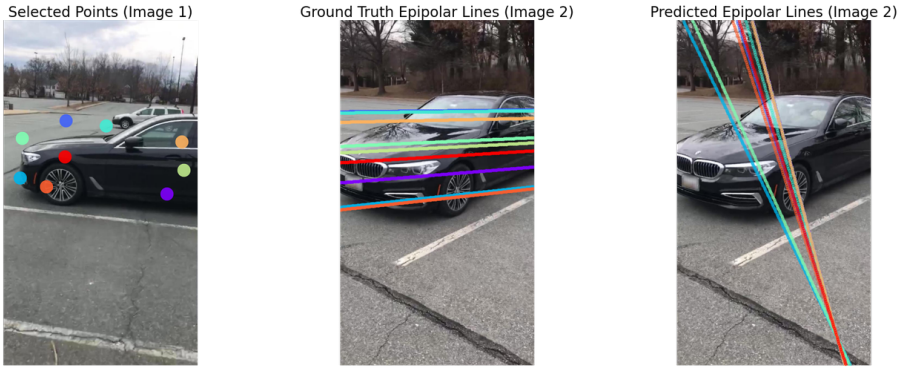


Figure 2.22: Failure cases from the SOP [Oh Song et al. 2016] dataset.



(a) Success case.



(b) Failure case.

Figure 2.23: Qualitative examples demonstrating the Epipolar geometry predicted using a pretrained LoFTR [J. Sun et al. 2021] for matching and MAGSAC++ [Barath et al. 2020] for robust optimization.

Chapter 3

Contrastive Lift: 3D Object Instance Segmentation by Slow-Fast Contrastive Fusion

The paper was published at the Neural Information Processing Systems (NeurIPS), 2023.

Contrastive Lift: 3D Object Instance Segmentation by Slow-Fast Contrastive Fusion

Yash Bhalgat Iro Laina João F. Henriques
Andrew Zisserman Andrea Vedaldi
Visual Geometry Group
University of Oxford
{yashsb,iro,joao,az,vedaldi}@robots.ox.ac.uk

May 4, 2026

Abstract

Instance segmentation in 3D is a challenging task due to the lack of large-scale annotated datasets. In this paper, we show that this task can be addressed effectively by leveraging instead 2D pre-trained models for instance segmentation. We propose a novel approach to lift 2D segments to 3D and fuse them by means of a neural field representation, which encourages multi-view consistency across frames. The core of our approach is a *slow-fast* clustering objective function, which is scalable and well-suited for scenes with a large number of objects. Unlike previous approaches, our method does not require an upper bound on the number of objects or object tracking across frames. To demonstrate the scalability of the slow-fast clustering, we create a new semi-realistic dataset called the Messy Rooms dataset, which features scenes with up to 500 objects per scene. Our approach outperforms the state-of-the-art on challenging scenes from the ScanNet, Hypersim, and Replica datasets, as well as on our newly created Messy Rooms dataset, demonstrating the effectiveness and scalability of our slow-fast clustering method.

3.1 Introduction

While the content of images is three-dimensional, image understanding has largely developed by treating images as two-dimensional patterns. This was primarily due to the lack of effective machine learning tools that could model content in 3D. However, recent advancements in neural field methods [Sitzmann et al. 2019; Mildenhall et al. 2020; Müller et al. 2022; A. Chen et al. 2022; Zhi et al. 2021a] have provided an effective approach for applying deep learning to 3D signals. These breakthroughs enable us to revisit image understanding tasks in 3D, accounting for factors such as multi-view consistency and occlusions.

In this paper, we study the problem of *object instance segmentation* in 3D. Our goal is to extend 2D instance segmentation to the third dimension, enabling simultaneous 3D reconstruction and 3D instance segmentation. Our approach is to extract information from multiple views of a scene independently with a pre-trained 2D instance segmentation model and fuse it into a single 3D neural field. Our main motivation is that, while acquiring densely labelled 3D datasets is challenging, annotations and pre-trained predictors for 2D data are widely available. Recent approaches have also capitalized on this idea, demonstrating their potential for 2D-to-3D semantic segmentation [Zhi et al. 2021a; Vora et al. 2021; Mascaro et al. 2021; Kundu et al. 2022] and distilling general-purpose 2D features in 3D space [Kobayashi et al. 2022a; Tschernezki et al. 2022]. When distilling semantic labels or features, the information to be fused is inherently consistent across multiple views: semantic labels are viewpoint invariant, and 2D features across views are typically learned with the same loss function. Additionally, the number of labels or feature dimensions is predetermined. Thus, 3D fusion amounts to multi-view aggregation.

When it comes to instance segmentation, however, the number of objects in a 3D scene is not fixed or known, and can indeed be quite large compared to the number of semantic classes. More importantly, when objects are detected independently in different views, they are assigned different and inconsistent identifiers, which cannot be aggregated directly. The challenge is thus how to fuse information that is not presented in a viewpoint-consistent manner.

Recently, Panoptic Lifting [Siddiqui et al. 2023a] proposed to resolve the lack

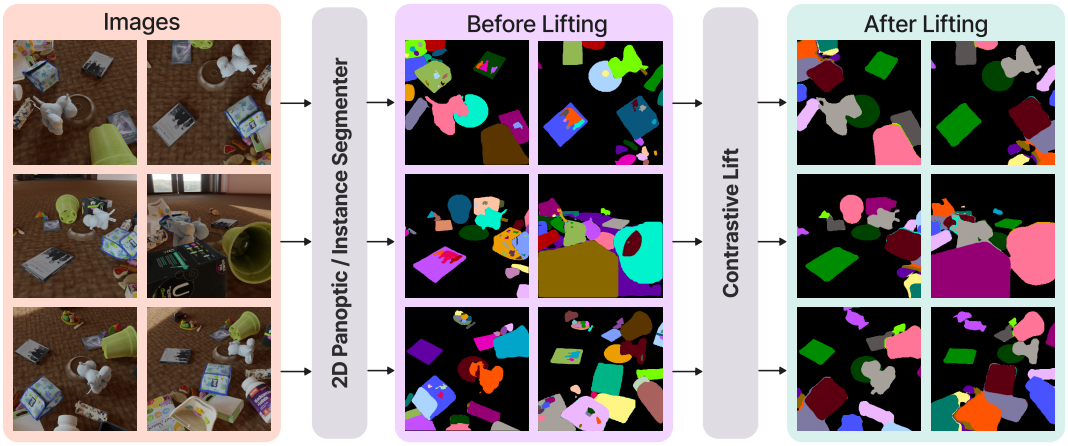


Figure 3.1: **Contrastive Lift** takes as input several views of a scene (left), as well as the output of a panoptic 2D segmenter (middle). It then reconstructs the scene in 3D while fusing the 2D segments, which are noisy and generally labelled inconsistently between views, when no object association (tracking) is assumed. Our method represents object instances in 3D space by a low-dimensional continuous embedding which can be trained efficiently using a contrastive formulation that is agnostic to the inconsistent labelling across views. The result (right) is a consistent 3D segmentation of the objects, which, once imaged, results in more accurate and consistent 2D segmentations.

of multi-view consistency by explicitly fitting a permutation that aligns labels extracted from multiple views. Although this yields good results, there are two drawbacks to this approach. Firstly, determining the permutation matrix involves solving a linear assignment problem using Hungarian Matching for every gradient computation. The cost of this increases cubically with the number of identifiers, which may limit scalability when dealing with a large number of object instances. Secondly, the canonical label space, where the permutation maps each 2D label, may need to be extensive to accommodate a large number of objects.

In this study, we propose a more efficient formulation, which also leads to more accurate results. To understand our approach, consider first a 2D image segmenter: it takes an image I as input and produces a mapping y that assigns each pixel $u \in \mathbb{R}^2$ to an object instance label $y(u) \in \{1, \dots, L\}$. It is natural to extend this mapping to 3D by introducing a function Y that associates each 3D point $x \in \mathbb{R}^3$ with the label $Y(x)$ of the corresponding object. To account for the fact that labels are arbitrary and thus inconsistent between views, Panoptic Lifting [Siddiqui et al. 2023a] seeks an image-dependent permutation matrix P such that $Y(x) = P \cdot y(u)$, where u is the projection of x onto the image.

To address the aforementioned challenges with the linear-assignment-based approach, we identify the labels $y(u)$ with coordinate vectors in the Euclidean space $\mathbb{R}^L$. The functions $y(u)$ can be reconstructed, up to a label permutation, from the distances $d(y(u), y(u')) = \|y(u) - y(u')\|_2$ of such vectors, as they tell whether labels of two pixels (u, u') are the same or different, without considering the specific labelling. Notably, similar to compressed sensing, we can seek lower-dimensional projections of the vectors y that preserve this information. With this in mind, we replace the 3D labelling function Y with a low-dimensional Euclidean embedding $\Theta(x) \in \mathbb{R}^D$. Then, we supervise the embeddings such that their distances $d(\Theta(x), \Theta(x')) \approx d(y(u), y(u'))$ are sufficiently similar to that of corresponding 2D label embeddings.

This approach has two advantages. First, it only requires learning vectors of dimensionality $D \ll L$ which is independent of the number of objects L. Second, learning this function does not require solving an assignment problem; rather, it only considers pairwise distances. Hence, the complexity of computing the learning objective is independent of the number of objects in the scene.

We translate this idea into a neural fusion field framework, which we call *Contrastive Lift*. We build on the recent progress in self-supervised learning, and combine two key ideas: the usage of a contrastive loss, and the usage of a slow-fast learning scheme for minimizing the latter in a stable manner. We believe to be the first to introduce these two ideas in the context of neural fields.

We compare our method to recent techniques including Panoptic Lifting [Siddiqui et al. 2023a] on standard 3D instance segmentation benchmarks, *viz.* ScanNet [Dai et al. 2017], Replica [Straub et al. 2019], and Hypersim [Roberts et al. 2021]. To better demonstrate the scalability of our method to a very large number of object instances, we introduce a semi-realistic Messy Rooms dataset featuring scenes with up to 500 objects.

3.2 Related Work

Neural Radiance Fields (NeRFs). NeRF [Mildenhall et al. 2020] and its numerous variants [A. Chen et al. 2022; Müller et al. 2022; Barron et al. 2021; Martin-Brualla et al. 2021; L. Liu et al. 2020] have achieved breakthrough results

in generating photorealistic 3D reconstructions from 2D images of a scene. These systems typically represent the scene as a continuous volumetric function that can be evaluated at any 3D point, enabling high-quality rendering of novel views from any viewpoint.

Objects and Semantics in NeRF. While NeRF by default offers low-level modelling of radiance and geometry, recent methods have expanded the set of tasks that can be addressed in this context to include *semantic* 3D modelling and scene decomposition. Some works use neural scene representations to decompose scenes into foreground and background without supervision or from weak signals [Z. Fan et al. 2023; C. Xie et al. 2021; H.-X. Yu et al. 2022; Tschernezki et al. 2021; Sharma et al. 2023; Mirzaei et al. 2022], such as text or object motion. Others exploit readily available annotations for 2D datasets to further extend the capabilities of NeRF models. For example, Semantic NeRF [Zhi et al. 2021a] proposes to incorporate a separate branch predicting semantic labels, while NeSF [Vora et al. 2021] predicts a semantic field by feeding a density field as input to a 3D semantic segmentation model.

Closer to our work are methods that employ NeRFs to address the problem of 3D panoptic segmentation [Kundu et al. 2022; Siddiqui et al. 2023a; Fu et al. 2022; WANG et al. 2023; Y. Liu et al. 2023]. Panoptic NeRF [Fu et al. 2022] and Instance-NeRF [Y. Liu et al. 2023] make use of 3D instance supervision. In Panoptic Neural Fields [Kundu et al. 2022], each instance is represented with its own MLP but dynamic object tracking is required prior to training the neural field. In this work, we focus on the problem of lifting 2D instance segmentation to 3D without requiring any 3D masks or object tracks. A paper most related to our work is Panoptic Lifting [Siddiqui et al. 2023a], which also seeks to solve the same problem, using linear assignment to make multi-view annotations consistent. Here, we propose a more efficient and effective technique based on learning permutation-invariant embedding vectors instead.

Fusion with NeRF. The aforementioned works, such as Semantic NeRF [Zhi et al. 2021a] or Panoptic Lifting [Siddiqui et al. 2023a], are also representative of a recent research direction that seeks to *fuse* the output of 2D analysis into 3D space. This is not a new idea; multi-view semantic fusion methods [Hermans et al.

2014; McCormac et al. 2017; Sünderhauf et al. 2017; L. Ma et al. 2017; Mascaro et al. 2021; Vineet et al. 2015] predate and extend beyond NeRF. The main idea is that multiple 2D semantic observations (e.g., noisy or partial) can be combined in 3D space and re-rendered to obtain clean and multi-view consistent labels. Instead of assuming a 3D model, others reconstruct a semantic map incrementally using SLAM [Kundu et al. 2014; Tateno et al. 2017; Narita et al. 2019]. Neural fields have greatly improved the potential of this idea. Instead of 2D labels, recent works, such as FFD [Kobayashi et al. 2022a], N3F [Tschernezki et al. 2022], and LERF [Kerr et al. 2023], apply the 3D fusion idea directly to supervised and unsupervised dense features; in this manner, unsupervised semantics can be transferred to 3D space, with benefits such as zero-shot 3D segmentation.

Slow-fast contrastive learning. Many self-supervised learning methods are based on the idea of learning representations that distinguish different samples, but are similar for different augmentations of the same sample. Some techniques build on InfoNCE [van den Oord et al. 2019; Tschannen et al. 2020] and, like MoCo [K. He et al. 2019] and SimCLR [Ting Chen et al. 2020], use a contrastive objective. Others such as SWaV [Caron et al. 2020] and DINO [Caron et al. 2021] are based on online pseudo-labelling. Many of these methods stabilise training by using mean-teachers [Tarvainen and Valpola 2017], also called momentum encoders [K. He et al. 2019]. The idea is to have two versions of the same network: a *fast* "student" network supervised by pseudo-labels generated from a *slow* "teacher" network, which is in turn updated as the moving average of the student model. Our formulation is inspired by this idea and extends it to learning neural fields.

Clustering operators for segmentation. Some works [Kong and Fowlkes 2018; Fathi et al. 2017; De Brabandere et al. 2017; Novotny et al. 2018] have explored using clustering of pixel-level embeddings to obtain instance segment assignments. Recent works [Q. Yu et al. 2022b; Q. Yu et al. 2022a] learn a pixel-cluster assignment by reformulating cross-attention from a clustering perspective. Our proposed method, Contrastive Lift, is similar in spirit, although we learn the embeddings (and cluster centers) using volumetric rendering from 2D labels.

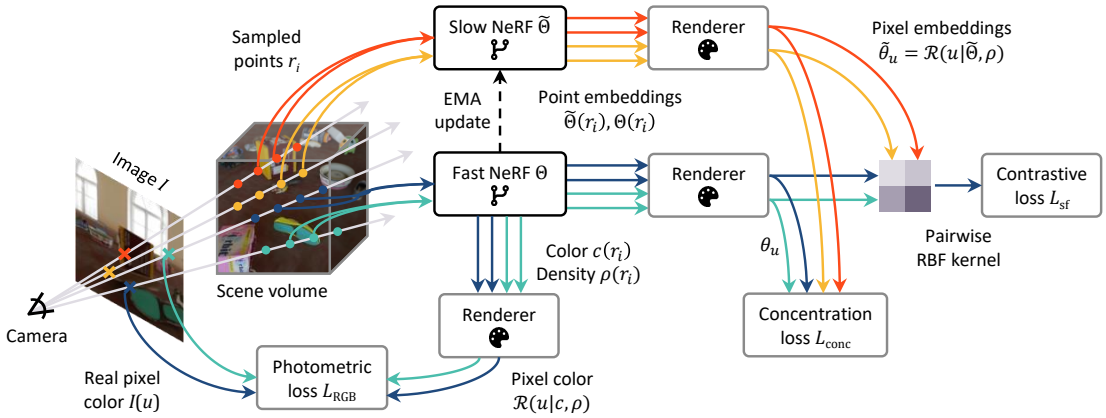


Figure 3.2: Overview of the Contrastive Lift architecture. See Section 3.3 for details.

3.3 Proposed Method: Contrastive Lift

Here and in Fig. 3.2, we describe Contrastive Lift, our approach for fusing 2D instance segmentation in 3D space. An image is a mapping $I : \Omega \to \mathbb{R}^3$, where Ω is a pixel grid in $\mathbb{R}^2$, and the values are RGB colours. We have a set of images $\mathcal{I}$ captured in the same scene and, for each image $I \in \mathcal{I}$, we have its camera pose $\pi \in SE(3)$ as well as object identity labels $y : \Omega \to \{1, \dots, L\}$ obtained from a 2D instance segmentation model for the image I. The labels y assigned to the 3D objects in one image I and the labels y' in another image I' are in general not consistent. Furthermore, these 2D label maps can be noisy across views.

We use this data to fit a neural field. The latter is a neural network that maps 3D coordinates $x \in \mathbb{R}^3$ to multiple quantities. The first two quantities are density, denoted by $\rho : \mathbb{R}^3 \mapsto [0, 1]$, and radiance (colour), denoted by $c : \mathbb{R}^3 \times \mathbb{S}^2 \mapsto [0, 1]^3$. Following the standard neural radiance field approach [Martin-Brualla et al. 2021], the colour $c(x, d)$ also depends on the viewing direction $d \in \mathbb{S}^2$. The third quantity is a D-dimensional instance embedding (vector) denoted as $\Theta : \mathbb{R}^3 \mapsto \mathbb{R}^D$. Each 3D coordinate is also mapped to a semantic embedding that represents a distribution over the semantic classes.

Differentiable rendering. The neural field associates attributes (density, colour, and embedding vectors) to each 3D point $x \in \mathbb{R}^3$. These attributes are projected onto an image I taken from a viewpoint π via differentiable ray casting. Given a pixel location $u \in \Omega$ in the image, we take N successive 3D samples $r_i \in \mathbb{R}^3$, $i = 0, \dots, N-1$ along the ray from the camera center through the

pixel (so that $(u, f) \propto \pi^{-1}(r_i)$ where f is the focal length). The probability that a photon is not absorbed when travelling from sample r_i to sample r_{i+1} is $\exp(-\rho(r_i)\delta_i)$ where $\delta_i = \|r_{i+1} - r_i\|_2$ is the distance between points. The *transmittance* $\tau_i = \exp(-\sum_{j=0}^{i-1} \rho(r_j)\delta_j)$ is the probability that the photon travels through sample r_i. The projection of any neural field $\mathbf{f}$ onto pixel u is thus given by the rendering equation:

$$\mathcal{R}(u|\mathbf{f}, \rho, \pi) = \sum_{i=0}^{N-1} \mathbf{f}(r_i)(\tau_i - \tau_{i+1}) = \sum_{i=0}^{N-1} \mathbf{f}(r_i)\tau_i(1 - \exp(-\rho(r_i)\delta_i)) \tag{3.1}$$

In particular, the colour of a pixel is reconstructed as $I(u) \approx \mathcal{R}(u|c(\cdot, d_u), \rho, \pi)$ where the viewing direction $d_u = r_0/\|r_0\|_2$. The photometric loss is thus:

$$\mathcal{L}_{\text{RGB}}(c, \rho|I) = \frac{1}{|\Omega|} \sum_{u \in \Omega} \|I(u) - \mathcal{R}(u|c(\cdot, d_u), \rho, \pi)\|^2. \tag{3.2}$$

Instance embeddings and slow-fast contrastive learning. The photometric loss (3.2) learns the colour and density fields (c, ρ) from the available 2D views $\mathcal{I}$. Now we turn to learning the instance embedding field $\Theta : \mathbb{R}^3 \mapsto \mathbb{R}^D$. As noted in Section 3.1, the goal of the embeddings is to capture the (binary) distances between pixel labels sufficiently well. By that, we mean that the segments can be recovered, modulo a permutation of their labels, by simply *clustering* the embeddings a posteriori.

We cast learning the embeddings as optimising the following contrastive loss function:

$$\mathcal{L}_{\text{contr}}(\Theta, \rho|y) = -\frac{1}{|\Omega|} \sum_{u \in \Omega} \log \frac{\sum_{u' \in \Omega} \mathbf{1}_{y(u)=y(u')} \exp(\text{sim}(\theta_u, \theta_{u'}; \gamma))}{\sum_{u' \in \Omega} \exp(\text{sim}(\theta_u, \theta_{u'}; \gamma))}, \tag{3.3}$$

where $\theta_u = \mathcal{R}(u|\Theta, \rho, \pi)$, $\mathbf{1}$ is the indicator function, and $\text{sim}(x, x'; \gamma) = \exp(-\gamma\|x - x'\|^2)$ is a Gaussian RBF kernel used to compute the similarity between embeddings in Euclidean space. Therefore, pixels that belong to the same segment are considered positive pairs, and their embeddings are brought closer, while the embeddings of pixels from different segments are pushed apart. It is worth emphasizing that, since the object identity labels obtained from the underlying 2D segmenter are not consistent *across* images, $\mathcal{L}_{\text{contr}}$ is only applied to positive and negative pixel pairs sampled from the *same* image.

While Eq. 3.3 is logically sound, we found it to result in gradients with high variance. To address this, we draw inspiration from momentum-teacher approaches [Hénaff et al. 2022; Caron et al. 2021; Bai et al. 2023] and define a *slowly-updated* instance embedding field $\tilde{\Theta}$, with parameters that are updated with an exponential moving average of the parameters of Θ, instead of gradient descent. With this, we reformulate Eq. 3.3 as:

$$\mathcal{L}_{\text{sf}}(\Theta, \rho | y, \tilde{\Theta}) = -\frac{1}{|\Omega_1|} \sum_{u \in \Omega_1} \log \frac{\sum_{u' \in \Omega_2} \mathbf{1}_{y(u)=y(u')} \exp(\text{sim}(\theta_u, \tilde{\theta}_{u'}; \gamma))}{\sum_{u' \in \Omega_2} \exp(\text{sim}(\theta_u, \tilde{\theta}_{u'}; \gamma))}, \tag{3.4}$$

where $\theta_u = \mathcal{R}(u|\Theta, \rho, \pi)$, and $\tilde{\theta}_{u'} = \mathcal{R}(u'|\tilde{\Theta}, \rho, \pi)$. Here, we randomly partition the pixels Ω into two non-overlapping sets Ω_1 and Ω_2, one for the "fast" embedding field Θ, and another for the "slow" field $\tilde{\Theta}$. This avoids the additional cost of predicting and rendering each pixel's embedding using both models, and allows the computational cost to remain the same as for Eq. 3.3.

Concentration loss. In order to further encourage the separation of the embedding vectors Θ and thus simplify the extraction of the objects via *a posteriori* clustering, we introduce a loss function that further encourages the embeddings to form concentrated clusters for each object:

$$\mathcal{L}_{\text{conc}}(\Theta, \rho | y, \tilde{\Theta}) = \frac{1}{|\Omega_1|} \sum_{u \in \Omega_1} \left\| \theta_u - \frac{\sum_{u' \in \Omega_2} \mathbf{1}_{y(u)=y(u')} \tilde{\theta}_{u'}}{\sum_{u' \in \Omega_2} \mathbf{1}_{y(u)=y(u')}} \right\|^2. \tag{3.5}$$

This loss computes a centroid (average) embedding as predicted by the "slow" field $\tilde{\Theta}$ and penalizes the squared error between each embedding (as predicted by the "fast" field Θ) and the corresponding centroid. While this loss reduces the variance of the clusters, it is not a sufficient training objective by itself as it does not encourage the separation of different clusters, as done by Eq. 3.3 and 3.4.

Semantic segmentation. For semantic segmentation, we follow the same approach as Semantic NeRF [Zhi et al. 2021a], learning additional embedding dimensions (one per semantic class), rendering labels in the same manner as Eq. 3.1, and using the cross-entropy loss for fitting the semantic field. Additionally, we also leverage the segment consistency loss introduced in [Siddiqui et al. 2023a] which encourages the predicted semantic classes to be consistent within an image

segment.

Architectural details. Our neural field architecture is based on TensoRF [A. Chen et al. 2022]. For the density, we use a single-channel grid whose values represent the scalar density field directly. For the colour, a multi-channel grid predicts an intermediate feature which is concatenated with the viewing direction and passed to a shallow 3-layer MLP to predict the radiance field. The viewing directions are encoded using a frequency encoding [Mildenhall et al. 2020; Vaswani et al. 2017]. For the instance embedding field Θ (and also the "slow" field $\tilde{\Theta}$ which has the exact same architecture as the "fast" field Θ), we use a shallow 5-layer MLP that predicts an embedding given an input 3D coordinate. The same architecture is used for the semantic field. We use raw 3D coordinates directly *without* a frequency encoding for the instance and semantic components. More details are provided in the supplementary material.

Rendering instance segmentation maps. After training is complete, we sample 10^5 pixels from 100 random viewpoints (not necessarily training views) and render the *fast* instance field Θ at these pixels using the corresponding viewpoint pose. The rendered $10^5 \times D$ embeddings are clustered using HDBSCAN [McInnes et al. 2017] to obtain centroids, which are cached. Now, for any novel view, the field Θ is rendered and for each pixel, the label of the centroid nearest to the rendered embedding is assigned.

3.4 Messy Rooms Dataset

In order to study the scalability of our method to scenes with a large number of objects, we generate a semi-realistic dataset using Kubric [Greff et al. 2022]. To generate a scene, we first spawn N realistically textured objects, randomly sampled from the Google Scanned Objects dataset [Downs et al. 2022], without any overlap. The objects are dropped from their spawned locations and a physics simulation is run for a few seconds until the objects settle in a natural arrangement. The static scene is rendered from M inward-facing camera viewpoints randomly sampled in a dome-shaped shell around the scene. Background, floor, and lighting are based on 360° HDRI textures from PolyHaven [Zaal et al. 2021] projected onto a dome.

Specifically, we create scenes with $N = 25, 50, 100$, and 500 objects. The number of viewpoints, M is set to $\min(1200, \lfloor 600 \times \sqrt{N/25} \rceil)$, and the rendered image resolution is 512×512. To ensure that the focus is on the added objects, we use background textures *old_room* and *large_corridor* from PolyHaven that do not contain any objects. A total of 8 scenes are generated. The use of realistic textures for objects and background environments makes them representative of real-world scenarios.

Additionally, we would like to maintain a consistent number of objects per image as we increase the total number of objects so that the performance of the 2D segmenter is not a factor in the final performance. Firstly, we ensure that the floor area of the scene scales proportionally with the number of objects, preventing objects from becoming densely packed. Secondly, the cameras move further away from the scene as its extent increases. To ensure that the same number of objects is visible in each image, regardless of the scene size, we adjust the focal length of the cameras accordingly, i.e. $f = 35.0 \times \sqrt{N/25}$, creating an effect similar to magnification. This approach ensures a comparable object distribution in each image, while enabling us to study the scalability of our method.

We render the instance IDs from each camera viewpoint to create ground-truth instance maps. These ground-truth instance IDs remain consistent (tracked) across views, as they are rendered from the same 3D scene representation.[1] Fig. 3.3 shows illustrative examples from the dataset, which we name *Messy Rooms*. For evaluation (Section 3.5), semantic maps are required. As there is a large variety of different object types in Kubric, there is no off-the-shelf detector that can classify all of these, and since we are interested in the instance segmentation problem, rather than the semantic classes, we simply lump all object types in a single "foreground" class, which focuses the evaluation on the quality of instance segmentation. More details about the dataset are provided in Appendix A.

[1] In all experiments, *tracked* ground-truth instance maps are used only for evaluation and not to train models.

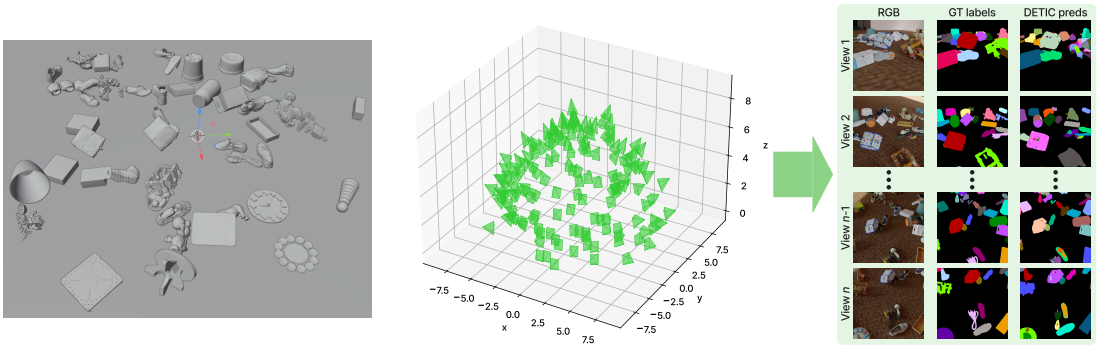


Figure 3.3: Messy Rooms dataset visualization. Left: physically realistic static 3D scene with N objects from GSO [Downs et al. 2022]. Middle: M camera viewpoints sampled in a dome-shaped shell. Right: ground-truth RGB and instance IDs, and instance segmentations obtained from Detic [X. Zhou et al. 2022].

3.5 Experiments

Benchmarks and baselines. We train and evaluate our proposed method on challenging scenes from the ScanNet [Dai et al. 2017], Hypersim [Roberts et al. 2021], and Replica [Straub et al. 2019] datasets. We compare our method with Panoptic Lifting (PanopLi) [Siddiqui et al. 2023a], which is the current state-of-the-art for lifting 2D panoptic predictions to 3D, along with other 3D panoptic segmentation approaches: Panoptic Neural Fields [Kundu et al. 2022] and DM-NeRF [WANG et al. 2023]. We follow PanopLi [Siddiqui et al. 2023a] for the data preprocessing steps and train-test splits for each scene from these datasets. We also evaluate our proposed method and PanopLi on our Messy Rooms dataset (Section 3.4) that features scenes with up to 500 objects. These experiments aim to demonstrate the scalability of our proposed method as compared to the linear-assignment approach.

We compare two variants of our Contrastive Lift method: (1) *Vanilla*: uses the simple contrastive loss (Eq. 3.3), and (2) *Slow-Fast*: uses slow-fast contrastive (Eq. 3.4) and concentration (Eq. 3.5) losses.

Metrics. The metric used in our evaluations is the scene-level Panoptic Quality (PQ^{scene}) metric introduced in [Siddiqui et al. 2023a]. PQ^{scene} is a scene-level extension of standard PQ [Kirillov et al. 2019] that takes into account the consistency of instance IDs across views/frames (*aka* tracking). In PQ^{scene}, predicted/ground-truth segments with the same instance ID across all views are merged into *subsets*

Table 3.1: Results on ScanNet, Hypersim, and Replica datasets. The performance of all prior work has been sourced from [Siddiqui et al. 2023a]. For each dataset, we report the PQ^{scene} metric.

Method	ScanNet [Dai et al. 2017]	HyperSim [Roberts et al. 2021]	Replica [Straub et al. 2019]
DM-NeRF [WANG et al. 2023]	41.7	51.6	44.1
PNF [Kundu et al. 2022]	48.3	44.8	41.1
PNF + GT BBoxes	54.3	47.6	52.5
PanopLi [Siddiqui et al. 2023a]	58.9	60.1	57.9
Vanilla (**Ours**)	60.5	60.9	57.8
Slow-Fast (**Ours**)	**62.3**	**62.3**	**59.1**

and all pairs of predicted/ground-truth *subsets* are compared, marking them as a match if the IoU is greater than 0.5.

Implementation Details. We train our neural field model for 400k iterations on all scenes. Optimization-related hyper-parameters can be found in Appendix B.2. The density grid is optimised using only the photometric loss ($\mathcal{L}_{\text{RGB}}$). While rendering the instance/semantic fields and computing associated losses (Eq. 3.3, 3.4, 3.5), gradients are stopped from flowing to the density grid.

For experiments on ScanNet, Hypersim and Replica, we use Mask2Former (M2F) [B. Cheng et al. 2022] as the 2D segmenter to obtain the image-level semantic labels and instance identities. Although any 2D segmenter can be used, using M2F allows direct comparisons with other state-of-the-art approaches [Siddiqui et al. 2023a]. We follow the protocol used in [Siddiqui et al. 2023a] to map the COCO [T.-Y. Lin et al. 2014] vocabulary to 21 classes in ScanNet.

For experiments on Messy Rooms, we use Detic [X. Zhou et al. 2022] instead since the object categories are not isomorphic to the COCO vocabulary M2F uses. We use the LVIS [Gupta et al. 2019] vocabulary with Detic. To show the scalability of our method compared to a linear-assignment-based approach, we train the PanopLi [Siddiqui et al. 2023a] model on this dataset. For fair comparison, we first train the density, colour and semantic fields, which are identical in PanopLi and our approach. We then separately train the instance field using the respective linear-assignment and slow-fast contrastive losses, with all other components frozen, ensuring that performance is only influenced by the quality of the learned instance field.

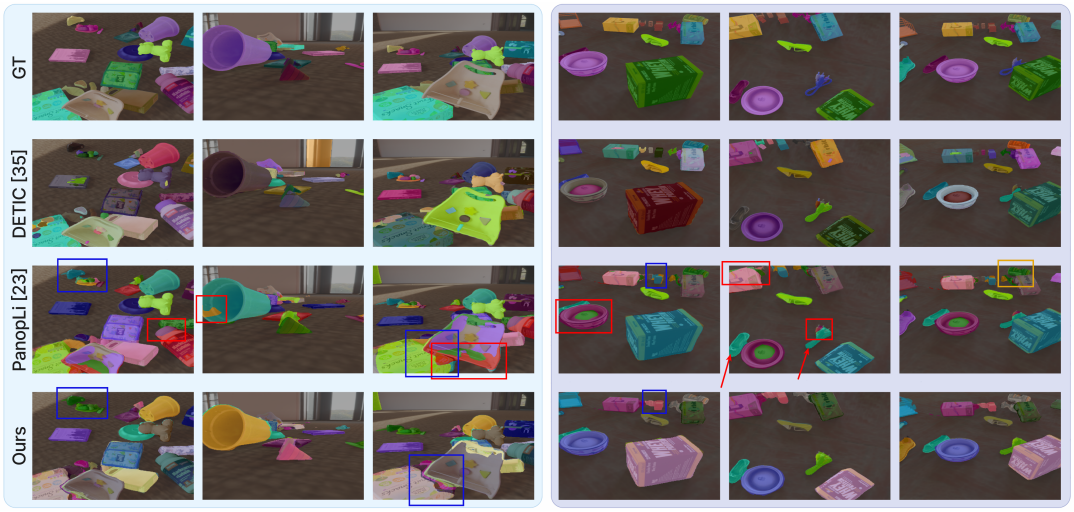
Table 3.2: Results on the Messy Rooms dataset. PQ^{scene} metric is reported on "old room" and "large corridor" environments with increasing number of objects in the scene ($N = 25, 50, 100, 500$).

Method	Old Room environment				Large Corridor environment			
	25 Objects	50 Objects	100 Objects	500 Objects	25 Objects	50 Objects	100 Objects	500 Objects
PanopLi [Siddiqui et al. 2023a]	73.2	69.9	64.3	51.0	65.5	71.0	61.8	49.0
Vanilla (**Ours**)	74.1	71.2	63.6	49.7	67.9	69.3	62.2	47.2
Slow-Fast (**Ours**)	**78.9**	**75.8**	**69.1**	**55.0**	**76.5**	**75.5**	**68.7**	**52.5**

3.5.1 Results

In Table 3.1, we compare the performance of our proposed approach with existing methods on three datasets: ScanNet [Dai et al. 2017], HyperSim [Roberts et al. 2021], and Replica [Straub et al. 2019]. Since the semantic field and underlying TensoRF [A. Chen et al. 2022] architecture we use is similar to Semantic-NeRF [Zhi et al. 2021a] and PanopLi [Siddiqui et al. 2023a], we only report the PQ^{scene} metric here and have added an additional table to Appendix D where we show that the mIoU and PSNR of our method match the performance of prior methods as expected. We observe that the proposed *Slow-Fast* approach consistently outperforms the baselines on all three datasets, while also outperforming the state-of-the-art Panoptic Lifting [Siddiqui et al. 2023a] method by +3.9, +1.4 and +0.8 PQ^{scene} points on these datasets respectively. We note that the *Vanilla* version of our method also performs comparably with PanopLi and outperforms other methods on all datasets.

Table 3.2 shows comparisons between our method and PanopLi [Siddiqui et al. 2023a] on scenes from our Messy Rooms dataset with $25, 50, 100$, and 500 objects. We see that the margin of improvement achieved by Contrastive Lift over PanopLi is even larger on these scenes, which shows that the proposed method scales favorably to scenes with a large number of objects. Fig. 3.4 shows qualitative results on two of these scenes. Even though the 2D segments obtained using Detic [X. Zhou et al. 2022] are noisy (sometimes *over-segmented*) and generally labelled inconsistently between views, the resulting instance segmentations rendered by Contrastive Lift are clearer and consistent across views. We also note that PanopLi sometimes fails to distinguish between distinct objects as pointed out in Fig. 3.4b.



(a) Messy Rooms: `large_corridor` (25 objects) (b) Messy Rooms: `old_room` (25 objects)

Figure 3.4: Qualitative comparisons of our method with PanopLi [Siddiqui et al. 2023a] and Detic [X. Zhou et al. 2022] (underlying 2D segmenter model) on scenes from our Messy Rooms dataset. **Colour coding:** regions where PanopLi performs poorly are highlighted with **red** boxes, while regions where both PanopLi and our method exhibit poor performance are marked with **blue** boxes. Additionally, **red** arrows indicate instances where PanopLi fails to distinguish between different objects. Please zoom in to observe finer details.

Table 3.3: Ablations of different variants of the Contrastive Lift method. PQ^{scene} metric averaged over the scenes of ScanNet and Messy Rooms datasets is reported. Embedding size of 3 is used.

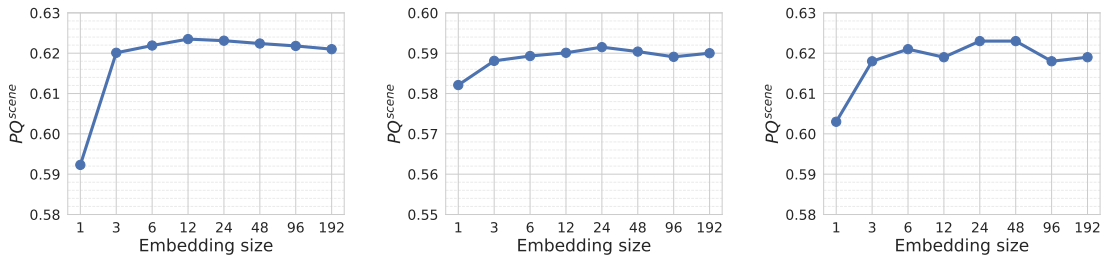
Dataset	$\mathcal{L}_{\text{sf}}+\mathcal{L}_{\text{conc}}$	$\mathcal{L}_{\text{sf}}$	$\mathcal{L}_{\text{contr}}$	$\mathcal{L}_{\text{contr}}+\mathcal{L}_{\text{conc}}$(fast)
ScanNet [Dai et al. 2017]	**62.0**	61.3	60.5	55.2
Messy Rooms	**69.0**	66.5	63.2	51.7

3.5.2 Ablations

Different variants of Contrastive Lift. Our proposed method uses $\mathcal{L}_{\text{sf}}$ (Eq. 3.4) and $\mathcal{L}_{\text{conc}}$ (Eq. 3.5) to optimise the instance embedding field. To study the effect of these losses, we design a comprehensive set of variants of the proposed method: **(1)** Proposed ($\underline{\mathcal{L}_{\text{sf}}+\mathcal{L}_{\text{conc}}}$), **(2)** Proposed without Concentration loss ($\underline{\mathcal{L}_{\text{sf}}}$), **(3)** Vanilla contrastive ($\underline{\mathcal{L}_{\text{contr}}}$), **(4)** Vanilla contrastive with Concentration loss applied to "fast" field since there is no "slow" field ($\underline{\mathcal{L}_{\text{contr}}+\mathcal{L}_{\text{conc}}(\text{fast})}$). Table 3.3 shows these ablations.

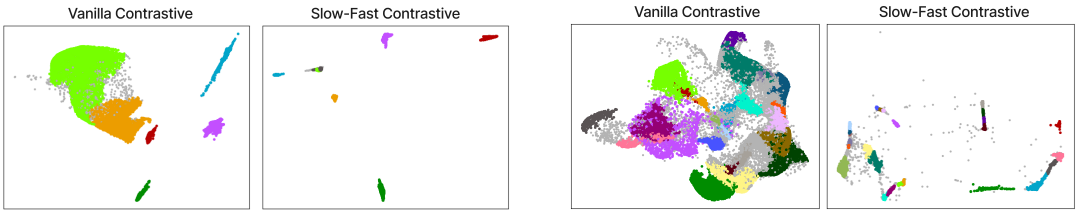
Effect of embedding size on performance. We investigate the impact of varying the instance embedding size on the performance of our proposed Contrastive Lift method. Specifically, we evaluate the effect of different embedding

sizes using the PQ^{scene} metric on ScanNet, Hypersim and Replica datasets. As shown in Fig. 3.5, we find that an embedding size as small as 3 is already almost optimal. Based on this, we use an embedding size of 24 for experiments with these datasets (*c.f.* Table 3.1). For experiments with Messy Rooms dataset (*c.f.* Table 3.2), we keep the embedding size to 3.



(a) ScanNet (b) Replica (c) Hypersim

Figure 3.5: Impact of the embedding size on the performance (PQ^{scene}) of the instance module.



(a) ScanNet: `scene0300_01` (b) Messy Rooms: `large_corridor_25`

Figure 3.6: Embeddings obtained using vanilla (plain) contrastive learning and our proposed Slow-Fast contrastive learning. We use LDA [Hastie et al. 2009] to project the embeddings to 2D for the illustration here.

Qualitative evaluation: Slow-fast vs vanilla contrastive learning. Fig. 3.6 shows how the embeddings are distributed in Euclidean space when learned using our proposed slow-fast contrastive loss (Eq. 3.4 and 3.5) and the vanilla contrastive loss (Eq. 3.3). Embeddings learned with the slow-fast method are clustered more compactly and are easy to distinguish using any post-processing algorithm, such as HDBSCAN [McInnes et al. 2017] which is used in this example.

Comparison to underlying 2D instance segmentation model with tracking. Before lifting, the predictions of the underlying 2D instance segmentation model (e.g., Mask2Former [B. Cheng et al. 2022] or Detic [X. Zhou et al. 2022]) are not consistent (*aka* tracked) across frames/views. To achieve consistency and

to allow comparisons with our approach, we post-process the 2D segmenter's predictions using Hungarian Matching for cross-frame tracking as follows:

1. **w/ Hungarian matching (2D IoU)**: Given sets of predicted segments (P_i and P_{i+1}) from consecutive frames, compute IoU matrix by comparing all segment pairs in $P_i \times P_{i+1}$. Apply Hungarian matching to the IoU matrix to associate instance segments across frames.

2. **w/ Hungarian matching based on IoU after depth-aware pose-warping**: Use ground-truth pose and depth for warping $(i+1)$-th frame's segmentation to frame i. Compute IoU matrix using warped segmentations and apply Hungarian matching.

3. **w/ Hungarian matching using ground-truth pointcloud**: Using only consecutive frames leads to errors in long-range tracking. To address this, starting from the first frame, unproject 2D segments into the 3D point cloud. Iteratively fuse segments in 3D using Hungarian matching. This way, segments from preceding frames along with 3D information are used for tracking.

The last two baselines use 3D groundtruth for tracking. Table 3.4 shows that despite 3D information being used for matching, Contrastive Lift still significantly improves over the underlying 2D model.

Table 3.4: Comparison of our approach with the underlying 2D segmentations on ScanNet [Dai et al. 2017]. For M2F predictions [B. Cheng et al. 2022], consistency across frames is obtained with different tracking variants.

Method	PQ^{scene}
Mask2Former [B. Cheng et al. 2022] (M2F) (**non-tracked**)	32.3
M2F w/ Tracking method (1)	33.7
M2F w/ Tracking method (2)	34.0
M2F w/ Tracking method (3)	41.0
Contrastive Lift (**ours** trained w/ Mask2Former labels)	**62.3**

Frame-level improvement on underlying 2D segmentation models. In addition to generating consistent (tracked) instance segmentations, our method also improves the per-frame quality (i.e., not considering tracking) of the underlying 2D segmentation model. To show this, we train Contrastive Lift on ScanNet scenes with different 2D models, *viz.* Mask2Former [B. Cheng et al. 2022], MaskFormer [B. Cheng et al. 2021] and Detic [X. Zhou et al. 2022]. In Table 3.5

we report the Panoptic Quality (PQ) metric (computed per frame) for these 2D models and for our method when trained with segments from each corresponding model.

Table 3.5: Improvement of per-frame segmentation quality as measured by Panoptic Quality (PQ).

Method	PQ
MaskFormer [B. Cheng et al. 2021]	41.1
Contrastive Lift (w/ MaskFormer labels)	61.7
Mask2Former [B. Cheng et al. 2022]	42.0
Contrastive Lift (w/ Mask2Former labels)	61.6
Detic [X. Zhou et al. 2022]	43.6
Contrastive Lift (w/ Detic labels)	**62.1**

Comparison of training speed with the linear-assignment loss method. While the exact number of objects present in a scene is unknown, linear assignment-based methods typically require a hyperparameter K that specifies the *maximum* number of objects. Solving the linear assignment problem in PanopLi's loss is $O(K^3)$ [Siddiqui et al. 2023a]. Our method is agnostic to object count, eliminating the need for such a parameter. Our approach does rely on the size of the embedding size, but, as shown above, even a very small size suffices. In the slow-fast contrastive loss computation, the Softmax function dominates more than the pairwise similarity matrix calculation. Consequently, we find that the training speed of Contrastive Lift is largely unaffected by the choice of embedding size.

Table 3.6 compares the training speed, measured on a NVIDIA A40 GPU, between PanopLi and our method, showing that PanopLi iterations become slower as K increases. We only optimise the instance embedding field with associated losses, while the density/colour/semantic fields are frozen.

Table 3.6: Training speed in iterations/second. Mean ± error margin measured over 8 runs.

Contrastive Lift	Panoptic Lifting [Siddiqui et al. 2023a]			
	$K = 25$	$K = 50$	$K = 100$	$K = 500$
16.06 ± 2.34	13.01 ± 1.26	12.53 ± 0.92	12.10 ± 1.07	9.41 ± 0.60

3.6 Limitations

Contrastive Lift improves noisy 2D input segmentations, but cannot recover from catastrophic failures, such as entirely missing object classes. It also requires the 3D reconstruction to work reliably. As a result, we have focused on static scenes, as 3D reconstruction remains unreliable in a dynamic setting. Contrastive Lift is a useful building block in applications, but has no particular direct societal impact. The datasets used in this paper are explicitly licensed for research and contain no personal data.

3.7 Conclusion

We have introduced Contrastive Lift, a method for fusing the outputs of 2D instance segmenter using a 3D neural fields. It learns a 3D vector field that characterises the different object instances in the scene. This field is fitted to the output of the 2D segmenter in a manner which is invariant to permutation of the object labels, which are assigned independently and arbitrarily in each input image. Compared to alternative approaches that explicitly seek to make multi-view labels compatible, Contrastive Lift is more accurate and scalable, enabling future work on larger object collections.

Acknowledgements. We are grateful for funding from EPSRC AIMS CDT EP/S024050/1 and AWS (Y. Bhalgat), ERC-CoG UNION 101001212 (A. Vedaldi and I. Laina), EPSRC VisualAI EP/T028572/1 (I. Laina, A. Vedaldi and A. Zisserman), and Royal Academy of Engineering RF\201819\18\163 (J. Henriques).

Data Ethics. We use the Google Scanned Objects, ScanNet, Hypersim and Replica datasets following their terms and conditions. These datasets do not contain personal data. For further details on ethics, data protection, and copyright please see https://www.robots.ox.ac.uk/~vedaldi/research/union/ethics.html.

3.8 Appendix

3.8.1 Messy Rooms dataset

The full Messy Rooms dataset introduced in this work can be accessed at this link: https://figshare.com/s/b195ce8bd8eafe79762b. We show some representative examples from this dataset below in Fig. 3.14, 3.15, 3.16, and 3.17, which illustrate scenes with $25, 50, 100$ and 500 objects respectively. Notice how the density of "number of objects per image" remains similar as the number of objects increases from 25 to 500. In Fig. 3.18, we show the corresponding 3D scenes used to generate the datasets.

3.8.2 Implementation Details

Here, we provide further implementation details for our method in addition to the details mentioned in Sections 3 and 5 of the main paper.

Architectural details

Our neural field architecture is similar to [Siddiqui et al. 2023a] for fairness of comparisons. The density and color grids are initialized with a resolution of 128× 128×128 which is progressively increased up to 192×192×192 by the end of training. The density and color grids use 16 and 48 components respectively. The output of the color grid is projected to 27 dimensions which are then processed by a 3-layer MLP with 128 hidden units per layer to output the RGB color. The *fast* and *slow* instance fields use a 256 hidden size in their MLP, while the semantic field uses a hidden size of 128.

Training details

We follow a schedule for training our neural field model as follows: (1) For the first 40k iterations, the model is trained only with the RGB reconstruction loss ($\mathcal{L}_{\text{RGB}}$). In this initial phase, the density field is optimized to reach a reasonable quality such that it can be used to render the instance/semantic field. (2) At 40k iterations, the semantic segmentation loss (i.e., cross-entropy loss, as in [Zhi et al. 2021a; Siddiqui et al. 2023a]) is activated and used for the rest of the training iterations. (3) At 160k iterations, the instance embedding loss (i.e., $\mathcal{L}_{\text{sf}} + \mathcal{L}_{\text{conc}}$ for the *slow*-

fast version of our method or $\mathcal{L}_{\text{contr}}$ for the *vanilla* baseline) is activated. (4) At 280k iterations, the segment consistency loss (proposed in [Siddiqui et al. 2023a]) is activated. For scenes from the Hypersim dataset [Roberts et al. 2021], we activate the segment consistency loss at 200k iterations instead. In our proposed slow-fast clustering framework, the *slow* field parameters are updated using an exponential moving average with momentum $m = 0.9$, i.e. $\tilde{\Theta} = \tilde{\Theta} \times m + \Theta \times (1 - m)$.

The RGB reconstruction loss, semantic segmentation loss, instance embedding loss, and segment consistency loss are balanced using weights of $1.0, 0.1, 0.1$, and 1.0 respectively. However, we empirically observe that the final performance is not very sensitive to these choices. A learning rate of $5 \cdot 10^{-4}$ is used for all MLPs and 0.01 for the grids. A batch-size of 2048 is used to train all models.

Post-processing Clustering details

Given the learned instance embedding field, a clustering mechanism (e.g., HDBSCAN [McInnes et al. 2017]) can be used to obtain cluster centroids and generate instance segmentation maps. We have chosen HDBSCAN since it does not require the number of objects to be known *a priori*. Generally, clusters obtained by HDBSCAN are non-convex and assigning the label of the nearest centroid is not recommended. But, our proposed method results in highly compact clusters which makes this simple method effective. We perform clustering as follows.

Hierarchical clustering using semantic predictions. An advantage of our method is that we can use our model's semantic predictions to guide the clustering of the instance embeddings. "Instance" segmentation requires separating instances of the same semantic class. Based on this, we perform hierarchical clustering as follows:

1. After training, sample 10^5 pixels from 100 random viewpoints and render the fast instance field Θ and semantic field for these pixels.
2. Group the 10^5 rendered embeddings based on predicted semantic labels, forming S groups.
3. Cluster the embeddings within each group separately using HDBSCAN, caching cluster-centroids for each group, assigning a unique instance label to each cen-

troid.

4. For a novel view, render the instance field and semantic field, assigning each pixel an instance embedding and semantic class. Obtain the instance label for a pixel by finding the closest centroid to the rendered instance embedding within the group of the same semantic label as the pixel.

Tuning clustering hyperparameter. Despite HDBSCAN's robustness to hyperparameter selection, we found that it is beneficial to specify a *minimum cluster size*. Since we always sample and render 10^5 pixels for clustering, the expected cluster size per object decreases as the number of objects increases. To determine an optimal value, we perform a hyperparameter sweep using 10% of the training data, which includes training viewpoints and associated segments from the 2D segmenter. We then use this identified optimal value to perform clustering as described above.

3.8.3 Comparison between different clustering algorithms.

We compare HDBSCAN with other unsupervised clustering algorithms, *viz.* MeanShift [Comaniciu and Meer 2002] and DBSCAN [Ester et al. 1996]. We tune the *bandwidth* parameter with MeanShift, and the *epsilon* parameter with DBSCAN. However, we note that MeanShift struggles to converge for embedding sizes greater than 10. For fair comparison, we train our model with an embedding size of 3. Table 3.7 show that both MeanShift and DBSCAN perform slightly worse but remain comparable to HDBSCAN. Generally, any unsupervised clustering method that doesn't require prior knowledge of the number of clusters is suitable for use with our method.

Table 3.7: Performance (PQ^{scene}) achieved with DBSCAN [Ester et al. 1996], MeanShift [Comaniciu and Meer 2002] and HDBSCAN [McInnes et al. 2017].

	ScanNet [Dai et al. 2017]	Messy Rooms
w/ DBSCAN	61.8	68.2
w/ MeanShift	62.0	68.6
w/ HDBSCAN	62.0	69.0

3.8.4 Quality of our semantic and radiance field

In Tables 1 and 2 in the main paper, we evaluate the quality and consistency (*aka* tracking) of the instance segmentation maps obtained by the various tested methods. The semantic field and density/color field architecture of our method is based on Panoptic Lifting [Siddiqui et al. 2023a], which in turn is a modification of Semantic-NeRF [Zhi et al. 2021a] for the semantic component. As a sanity check, we compare the quality of rendered semantic and RGB maps obtained by these methods with ours. Table 3.8 shows the mean Intersection over Union (mIoU) and peak-signal-to-noise ratio (PSNR) metrics. As expected, the mIoU and PSNR obtained by our method is nearly the same as Panoptic Lifting.

For the Messy Rooms dataset we have explicitly ensured that the density and semantic model used by both Panoptic Lifting and our method are the same and the only factor influencing the final performance is the quality of the learned instance field. This is done by pre-training the same density and semantic fields for both methods and subsequently training the instance field using the respective objective functions.

Table 3.8: Comparisons of the rendered semantic and RGB maps. Performance numbers for [Zhi et al. 2021a; Siddiqui et al. 2023a] are sourced from [Siddiqui et al. 2023a].

Method	ScanNet [Dai et al. 2017]		HyperSim [Roberts et al. 2021]		Replica [Straub et al. 2019]	
	mIoU	PSNR	mIoU	PSNR	mIoU	PSNR
Semantic-NeRF [Zhi et al. 2021a]	58.9	26.6	58.5	24.8	59.2	26.6
PanopLi [Siddiqui et al. 2023a]	**65.2**	**28.5**	67.8	**30.1**	**67.2**	29.6
Ours	**65.2**	28.3	**67.9**	30.0	67.0	29.3

3.8.5 Comparisons to other metric learning loss functions

While we employ a contrastive loss formulation to learn the instance embeddings, there are many alternative loss functions proposed in the metric learning literature. For comparison, we also train our instance embedding field with the Associative Embedding (AE) loss [Newell et al. 2017] and the margin-based contrastive loss [Chopra et al. 2005].

To compute the AE Loss, we divide the batch Ω into groups based on segment ID.

If there are K groups/segments, $G_1 \dots G_K$, then

$$\mathcal{L}_{\text{AE}}(\Theta, \rho|y) = \frac{1}{|\Omega|} \sum_k \sum_{u \in G_k} \|\theta_u - \bar{\theta}_k\|_2^2 + \frac{1}{K^2} \sum_k \sum_{k'} \|\bar{\theta}_k - \bar{\theta}_{k'}\|_2^2, \quad \bar{\theta}_k = \frac{1}{|G_k|} \sum_{u \in G_k} \theta_u \tag{3.6}$$

The margin-based contrastive loss is defined as:

$$\mathcal{L}_{\text{margin}}(\Theta, \rho|y) = \frac{1}{|\Omega|^2} \sum_{u,u' \in \Omega} \mathbf{1}_{[y(u)=y(u')]} \|\theta_u - \theta_{u'}\|_2^2 + \mathbf{1}_{[y(u)\neq y(u')]} \max(0, \epsilon - \|\theta_u - \theta_{u'}\|_2^2) \tag{3.7}$$

Here, $\theta_u = \mathcal{R}(u|\Theta, \rho, \pi)$ is the rendered instance field at pixel u. Note that, the slow-fast field formulation is not used in these comparisons. In Table 3.9 we compare the proposed objective (slow-fast) to these baselines. We observe that both the vanilla contrastive, as well as the slow-fast version of our method, outperform the alternatives.

DINO-style loss. Since our method is inspired by momentum-teacher approaches, e.g. DINO [Caron et al. 2021], we design a baseline with a DINO-style learning mechanism. Two pixels from the same instance segment are fed into the *slow* and *fast* fields. This is akin to DINO, where two random image transformations are fed to the student and teacher networks. A Centering layer is applied to the *slow* field embedding. A Projection module (with proj_dim = 512) is added to both the *slow* and *fast* fields followed by Softmax and a cross-entropy loss is used. After training, embeddings from the *fast* field are used for clustering. Results in Table 3.9 demonstrate that this baseline performs worse than the metric-learning losses on ScanNet. We do not evaluate this baseline on the Messy Rooms dataset.

Table 3.9: Comparing Associative Embedding Loss and Triplet Loss with our proposed losses (*vanilla* and *slow-fast*). An embedding size of 3 is used in all cases. PQ^{scene} is reported here.

	ScanNet [Dai et al. 2017]	Messy Rooms
Panoptic Lifting [Siddiqui et al. 2023a]	58.9	63.2
Ours w/ AE loss ($\mathcal{L}_{\text{AE}}$)	60.0	62.4
Ours w/ Margin loss ($\mathcal{L}_{\text{margin}}$)	60.1	62.9
Ours w/ DINO-style loss	54.7	-
Ours w/ Vanilla contrastive loss ($\mathcal{L}_{\text{contr}}$)	60.5	63.1
Ours w/ Slow-Fast losses (**proposed**)	**62.0**	**69.0**

3.8.6 Stability of Slow-Fast loss compared to Vanilla contrastive loss

We found that training with the vanilla contrastive loss resulted in gradients with higher variance. The usage of a slowly-updated embedding field in the slow-fast loss formulation mitigates this problem and leads to more stable training. We quantitatively verify this by computing the *relative variance* (which is $^{Var(\cdot)}/_{Mean(\cdot)}$) in the gradients of the loss w.r.t. to the instance embeddings (i.e. $^{dL}/_{d\Theta}$). Figure 3.7 shows that the vanilla loss exhibits spikes with a maximum relative variance around 10^7, whereas the slow-fast version remains around a much controlled range of around 10^1.

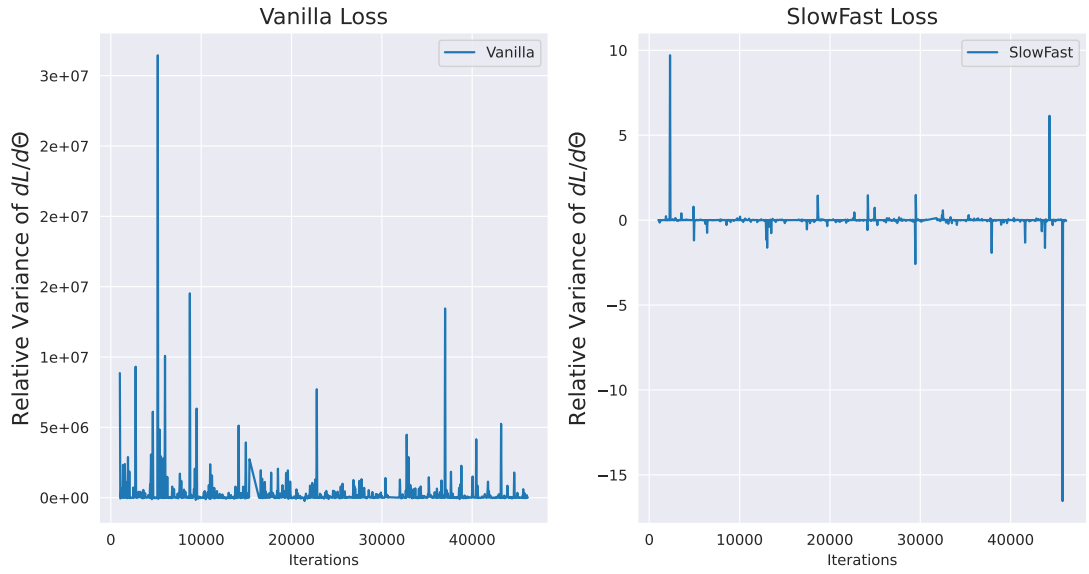


Figure 3.7: Relative variance (i.e. $^{Var(\cdot)}/_{Mean(\cdot)}$) in loss gradients w.r.t. embeddings (i.e. $^{dL}/_{d\Theta}$).

3.8.7 More qualitative visualizations

In Figures 3.8, 3.9, 3.10, 3.11, 3.12 and 3.13, we visualize the predictions of our proposed method on scenes from ScanNet [Dai et al. 2017] and Messy Rooms. Left-most columns show instance labels obtained after clustering, which as we can see are consistent across different views. To understand how well the embeddings are clustered, we visualize heatmaps of distance of rendered embeddings from cluster-centroids. Specifically, we choose 4 centroids, and for each centroid c_i and each pixel u, we plot $H(u) = -\log(\|\theta_u - c_i\|)$ normalized to $[0, 1]$, where θ_u is the

rendered embedding.

Note that, instance labels are only computed for pixels belonging to the "*thing*" semantic categories (as predicted by the semantic field)[2]. The "*stuff*" pixels are masked out.

As can be seen in all these visualizations, the heatmaps are peaked at the corresponding object locations and close to zero elsewhere which indicates the embeddings are compactly clustered around the corresponding centroid for each object. In Fig. 3.13, we can see that even in a scene with 50 objects, the embeddings for each instance are distinctly separable.

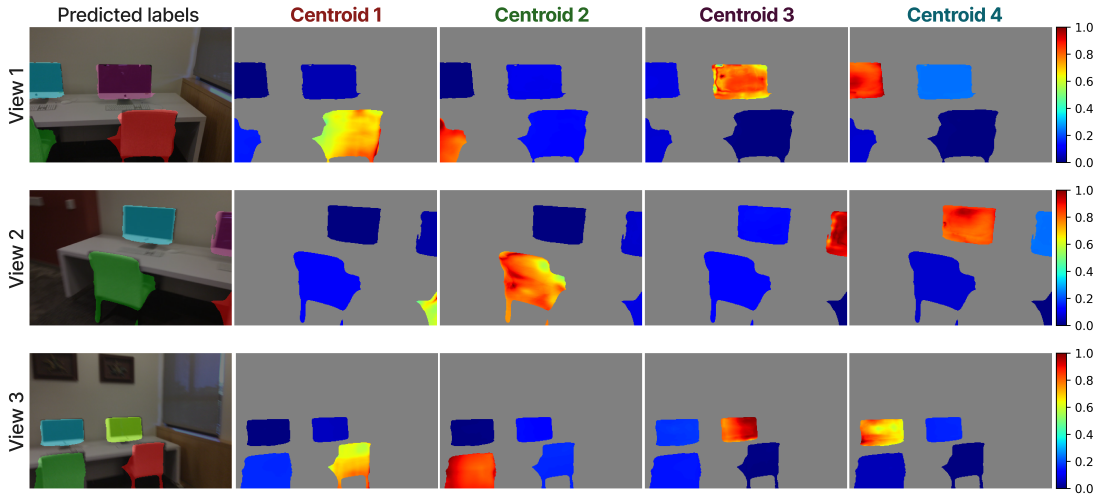


Figure 3.8: ScanNet scene0300_01: Visualized instance segmentation and clustering heatmaps.

[2]The thing and stuff categories for our Messy Rooms dataset are simply "foreground" and "background".

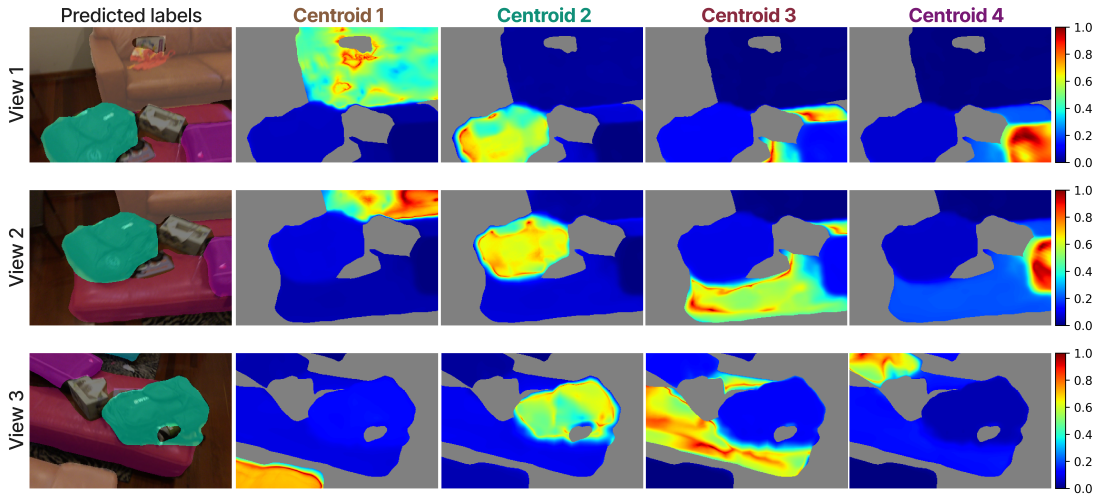


Figure 3.9: ScanNet scene0050_02: Visualized instance segmentation and clustering heatmaps.

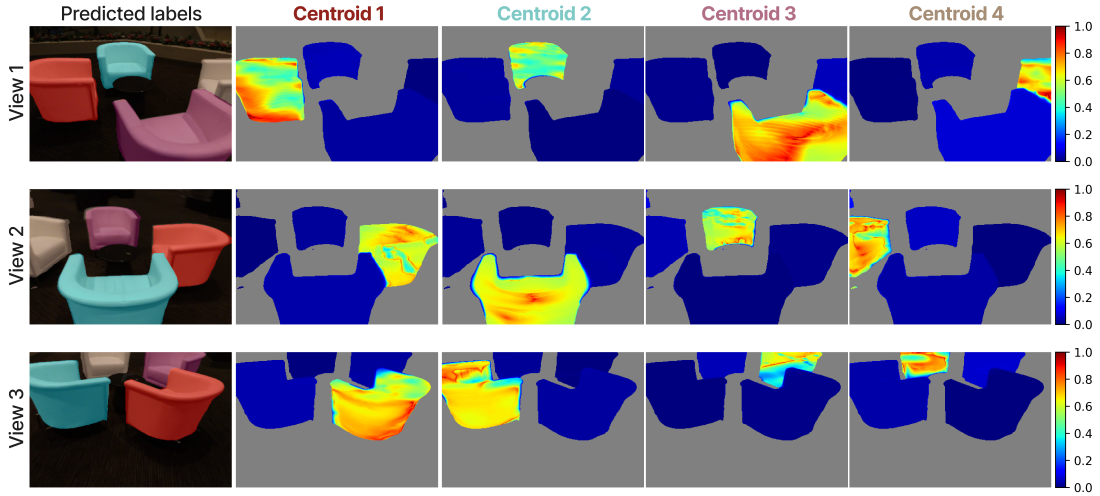


Figure 3.10: ScanNet scene0423_02: Visualized instance segmentation and clustering heatmaps.

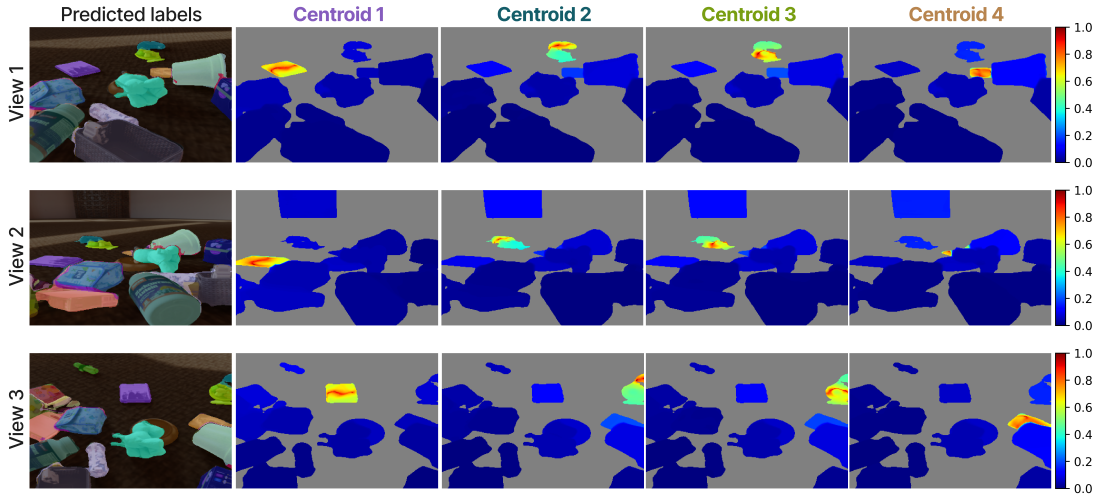


Figure 3.11: Messy Rooms large_corridor_25: Visualized instance segmentation and clustering heatmaps.

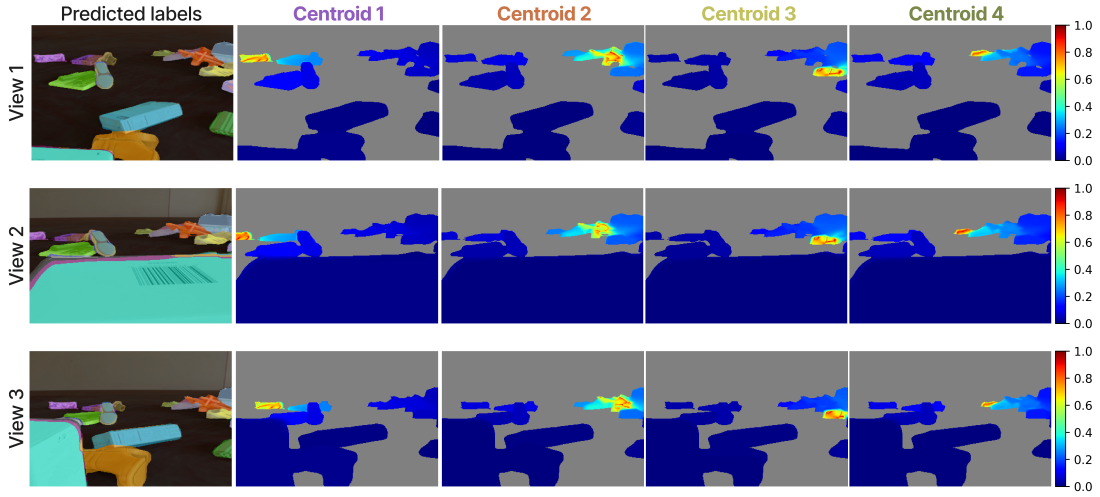


Figure 3.12: Messy Rooms old_room_25: Visualized instance segmentation and clustering heatmaps.

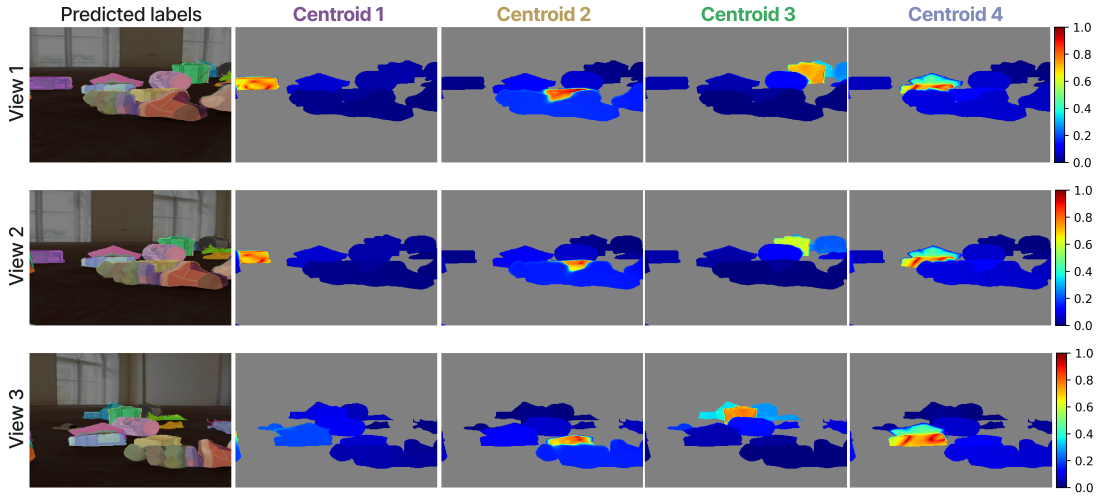
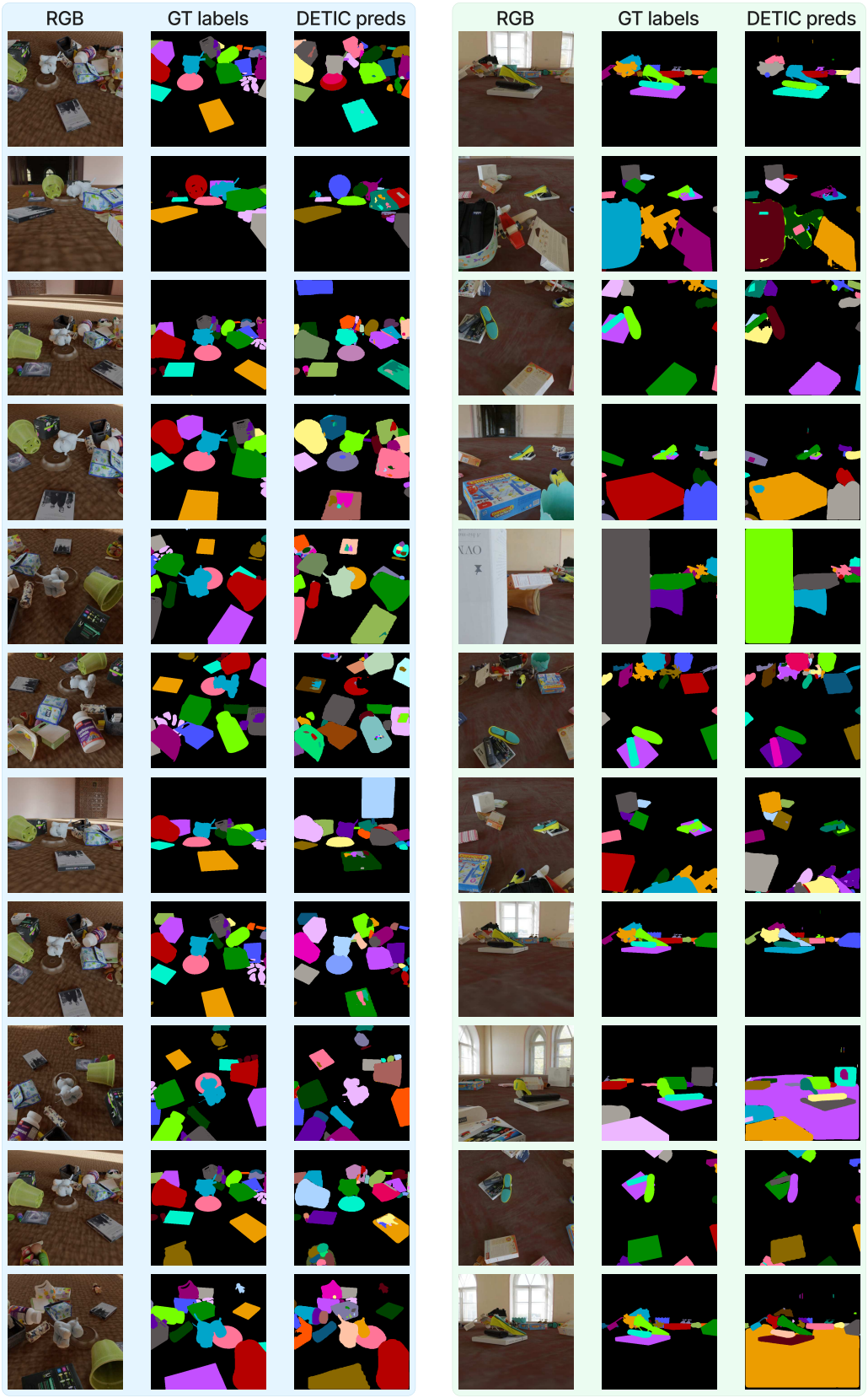
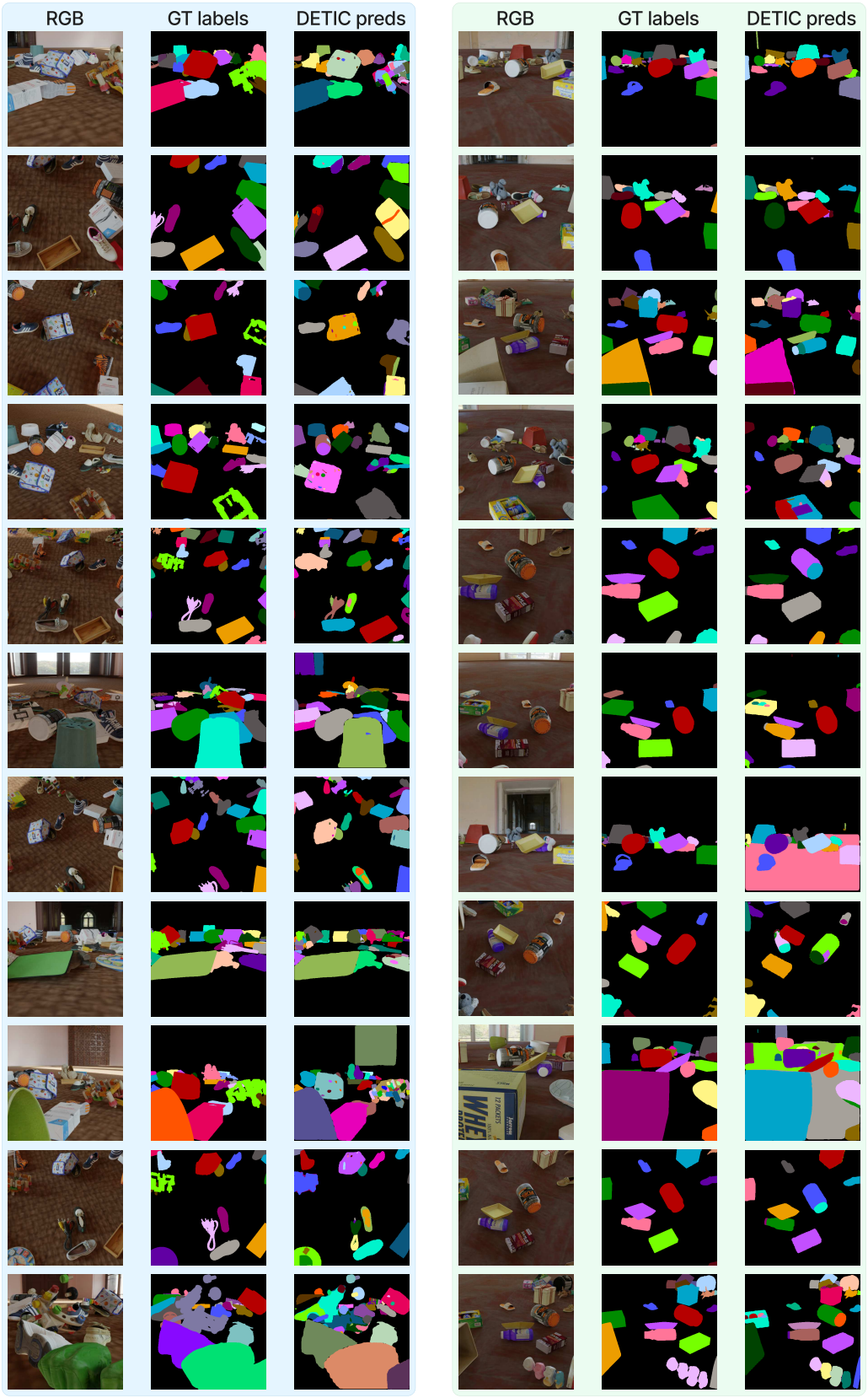


Figure 3.13: Messy Rooms old_room_50: Visualized instance segmentation and clustering heatmaps.



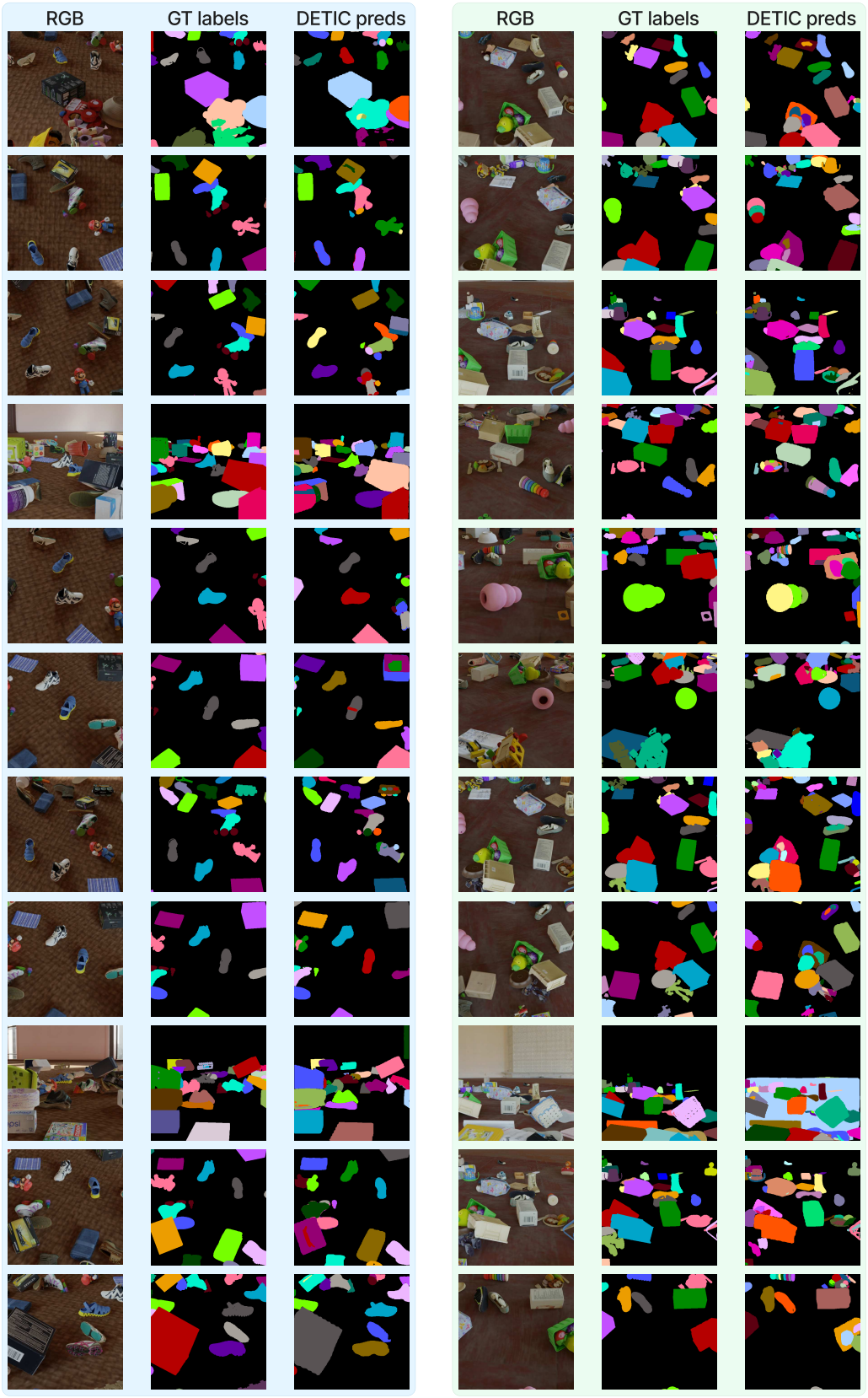
(a) `large_corridor`: 25 objects (b) `old_room`: 25 objects

Figure 3.14: Illustrative examples from Messy Rooms dataset. Here, we show scenes with 25 objects.



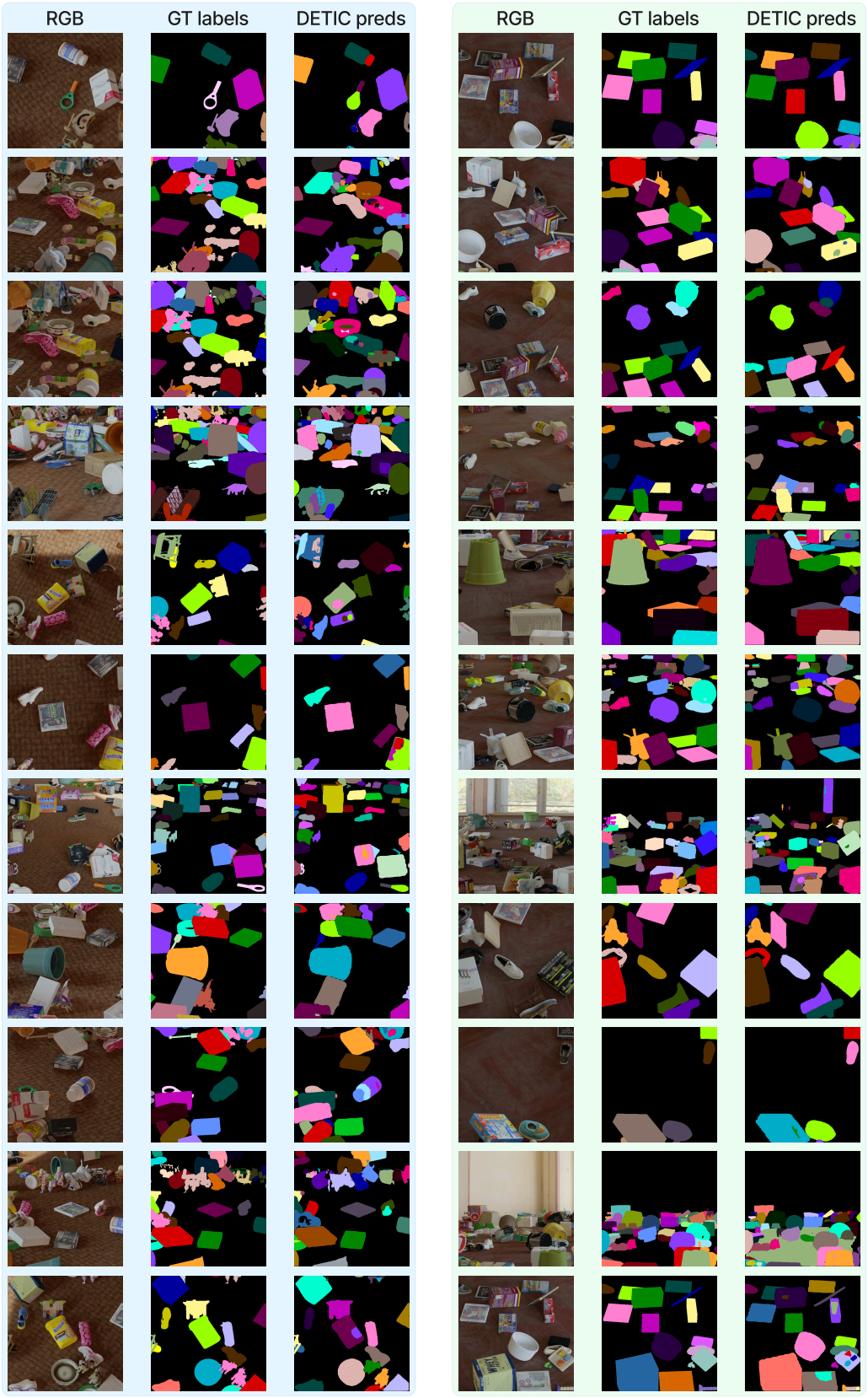
(a) `large_corridor`: 50 objects (b) `old_room`: 50 objects

Figure 3.15: Illustrative examples from Messy Rooms dataset. Here, we show scenes with 50 objects.



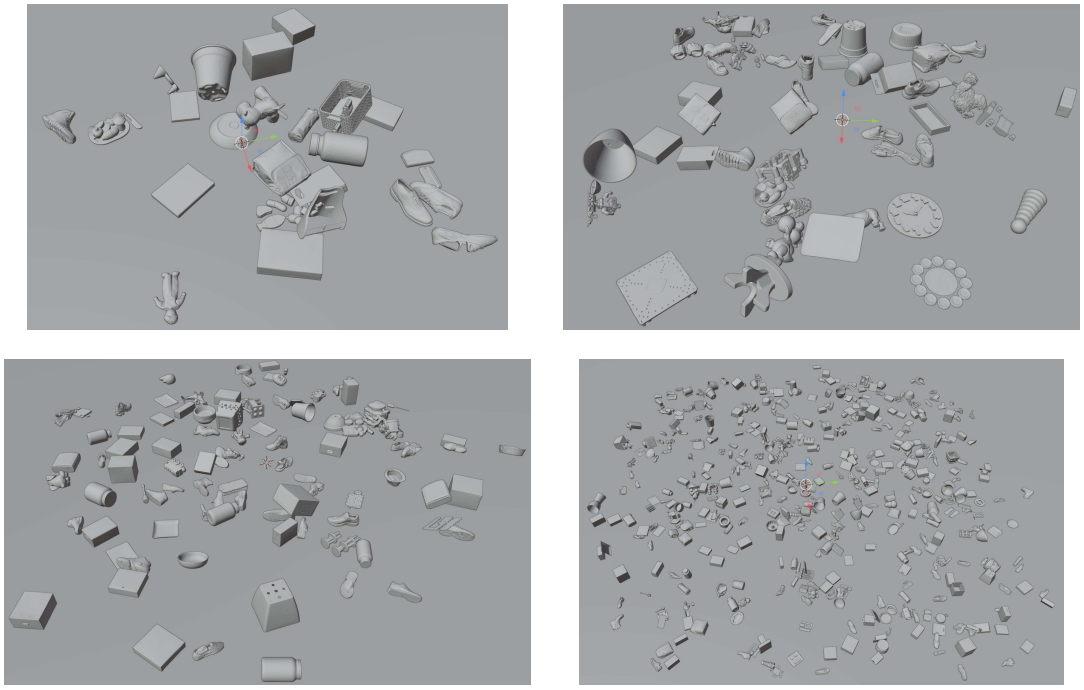
(a) `large_corridor`: 100 objects (b) `old_room`: 100 objects

Figure 3.16: Illustrative examples from Messy Rooms dataset. Here, we show scenes with 100 objects.

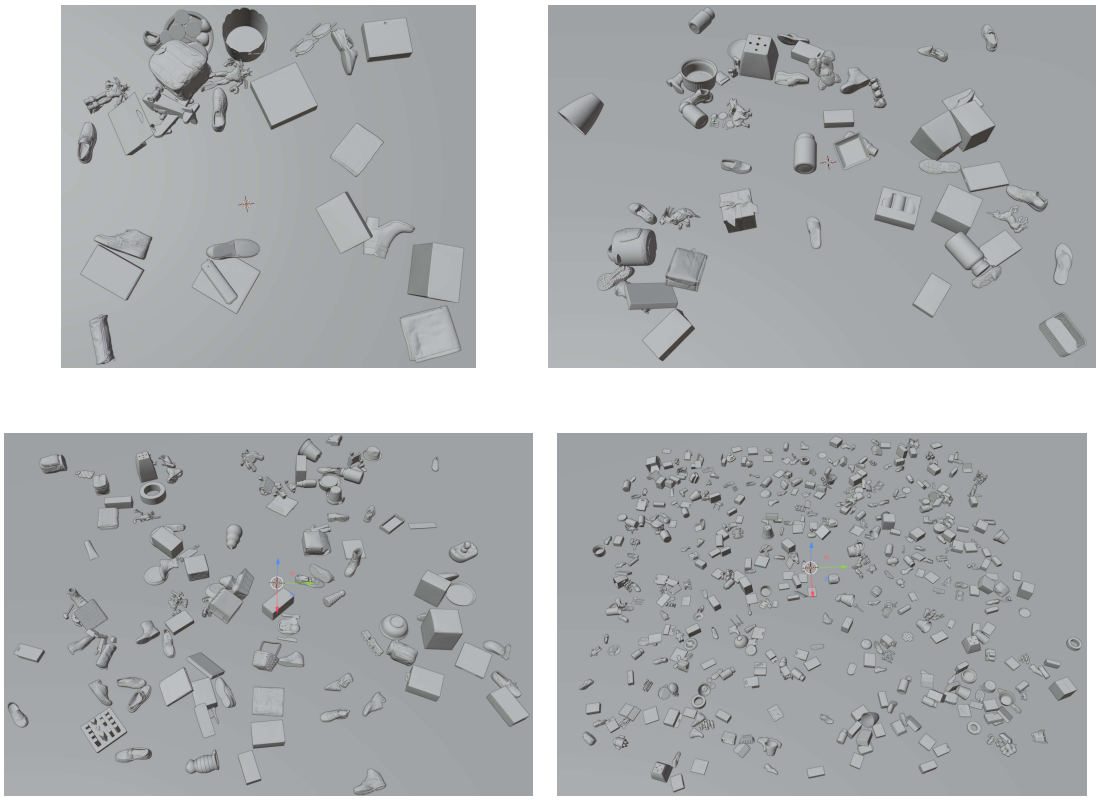


(a) `large_corridor`: 500 objects (b) `old_room`: 500 objects

Figure 3.17: Illustrative examples from Messy Rooms dataset. Here, we show scenes with 500 objects.



(a) Scenes with `large_corridor` environment. Top left: 25 objects. Top right: 50 objects. Bottom left: 100 objects. Bottom right: 500 objects.



(b) Scenes with `old_room` environment. Top left: 25 objects. Top right: 50 objects. Bottom left: 100 objects. Bottom right: 500 objects.

Figure 3.18: We show (using Blender [Community 2018]) the actual 3D scenes *without texture* that are used to render/generate the Messy Rooms scenes. *Note that the surface area of the scene is increased proportionally to the number of objects.*

Chapter 4

N2F2: Hierarchical Scene Understanding with Nested Neural Feature Fields

The paper was published at the European Conference on Computer Vision (ECCV), 2024.

N2F2: Hierarchical Scene Understanding with Nested Neural Feature Fields

Yash Bhalgat Iro Laina João F. Henriques
Andrew Zisserman Andrea Vedaldi
Visual Geometry Group
University of Oxford
{yashsb,iro,joao,az,vedaldi}@robots.ox.ac.uk

May 4, 2026

Abstract

Understanding complex scenes at multiple levels of abstraction remains a formidable challenge in computer vision. To address this, we introduce Nested Neural Feature Fields (N2F2), a novel approach that employs hierarchical supervision to learn a *single* feature field, wherein different dimensions within the same high-dimensional feature encode scene properties at varying granularities. Our method allows for a flexible definition of hierarchies, tailored to either the physical dimensions or semantics or *both*, thereby enabling a comprehensive and nuanced understanding of scenes. We leverage a 2D class-agnostic segmentation model to provide semantically meaningful pixel groupings at arbitrary scales in the image space, and query the CLIP vision-encoder to obtain language-aligned embeddings for each of these segments. Our proposed hierarchical supervision method then assigns different nested dimensions of the feature field to distill the CLIP embeddings using deferred volumetric rendering at varying physical scales, creating a coarse-to-fine representation. Extensive experiments show that our approach outperforms the state-of-the-art feature field distillation methods on tasks such as open-vocabulary 3D segmentation and localization, demonstrating the effectiveness of the learned nested feature field.

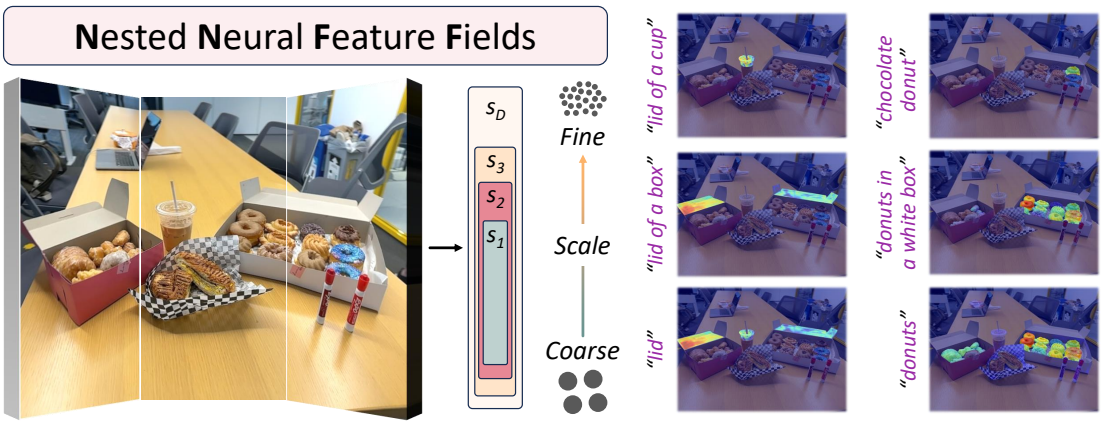


Figure 4.1: **Nested Neural Feature Fields (N2F2)**. We present N2F2, wherein different dimensions of the same feature field encode scene properties at varying granularities. The illustration captures the essence of hierarchical scene understanding, depicting how our model differentiates between *coarse* and *fine* scales to accurately interpret complex semantic queries, such as *"donuts in a white box"* and *"chocolate donut"*, showcasing the model's versatility in handling detailed object descriptions within 3D environments.

4.1 Introduction

3D scene understanding is an important problem in computer vision which still presents several challenges. One of them is that scene understanding is inherently hierarchical, as it requires reasoning about the scene at varying levels of geometric and semantic granularity. Models must simultaneously understand the high-level structure and composition of the scene as well as fine-grained object details. This is important for applications like robotics and augmented reality.

Recent progress in radiance fields has played a pivotal role in advancing 3D scene understanding. Methods such as Neural Radiance Fields (NeRF) [Mildenhall et al. 2020] and 3D Gaussian Splatting [Kerbl et al. 2023] can extract the shape and appearance of 3D scenes without 3D supervision or specialized sensors, as they are inferred directly from several RGB images via differentiable rendering. Furthermore, this optimization process can be extended to distill and fuse 2D information into the 3D representation well beyond RGB values. Several authors have in fact proposed *fusion approaches*, where 2D labels [Zhi et al. 2021b; Bhalgat et al. 2023; C. M. Kim et al. 2024a; Siddiqui et al. 2023b] or 2D features [Tschernezki et al. 2022; Kobayashi et al. 2022a; Kerr et al. 2023] are extracted from multiple views of the scene and fused into a single 3D model. The fused features not only augment the 3D reconstruction with a semantic interpretation, but can also remove noise

from the 2D labels, improving their quality.

In this work, we focus on the problem of 3D distillation of vision-language representations, such as CLIP [Radford et al. 2021], which, in turn, enables open-vocabulary 3D segmentation and localization of objects or scene elements based on natural language descriptions. Existing methods like LERF [Kerr et al. 2023] have demonstrated the potential of embedding language features into NeRFs, allowing users to query 3D scenes with arbitrary text inputs, but face two key limitations.

First, while vision-language models exhibit remarkable few-shot transfer capabilities, their performance often degrades for more complex linguistic constructs like compound nouns ("paper napkin") or partitive phrases ("bag of cookies"). This is because individual components may be interpreted separately (e.g., causing to detect "paper" and "napkin" as separate concepts instead of a "paper napkin"). This stems from the inherent challenge of compositional generalization that plagues such models and causes them to behave like bags-of-words [Yuksekgonul et al. 2022; Thrush et al. 2022; Z. Ma et al. 2023; Zhiqiu Lin et al. 2023]. This behavior makes it difficult to use vision-language models to capture compositions of objects and their attributes or relations and is often attributed to the text encoder bottleneck [Kamath et al. 2023] and the contrastive formulation that is used to train such models. In practice, this means that querying CLIP with the aforementioned prompts would falsely localize both paper *and* napkins, bags *and* cookies. Existing 3D feature distillation methods directly inherit these shortcomings, failing to accurately localize or segment objects described by compound expressions.

A second limitation of methods like LERF is their inefficiency during inference. To produce relevance maps for a given text query, these approaches densely evaluate the feature field at multiple spatial scales. This imposes a noticeable computational burden that grows linearly with the number of scales processed.

This work aims to address both limitations. Our key insight is to impose a *hierarchical structure* on the 3D feature field during training. Specifically, we propose **Nested Neural Feature Fields (N2F2)**, wherein different subsets of dimensions within a single high-dimensional feature field are tasked with encoding scene properties at varying granularities. This design choice enables N2F2 to simultaneously capture multi-scale scene representations in a coherent and parameter-

efficient manner. To address the compositionality challenge, we further contribute a novel composite embedding approach that, during inference, combines the features across all hierarchy levels in a weighted manner, effectively aggregating relevance scores across multiple scales given a text query and eliminating the need for explicit scale selection.

Through extensive experimentation on challenging datasets, we demonstrate that N2F2 significantly outperforms prior work on open-vocabulary 3D segmentation and localization, including those involving complex compound queries. Moreover, our composite embedding strategy yields considerable speedups during inference, making N2F2 1.7× faster than the current leading approach, LangSplat [Qin et al. 2023], with better accuracy and a significant increase in granularity.

In summary, our core contributions are: (1) A hierarchical supervision framework for distilling multi-scale semantic representations into a unified 3D feature field; (2) a composite embedding strategy that enables efficient open-vocabulary querying without explicit scale selection; (3) state-of-the-art performance on challenging open-vocabulary 3D segmentation and localization benchmarks, with particular gains on complex compound queries.

4.2 Related Work

Radiance Fields. Radiance fields have emerged as a powerful representation for capturing and rendering complex 3D scenes from a set of multi-view posed images. Neural Radiance Fields (NeRFs) [Mildenhall et al. 2020] pioneered this approach by representing a scene as a continuous 5D function, mapping spatial coordinates and viewing directions to colors and densities. These representations can be optimized by leveraging differentiable volume rendering. This seminal work has inspired a plethora of variants and extensions, such as deformable or dynamic NeRFs [Pumarola et al. 2020], NeRFs for unconstrained images [Martin-Brualla et al. 2021], and variants for improved efficiency [A. Chen et al. 2022; Chan et al. 2022; C. Sun et al. 2022]. Recent work on 3D Gaussian Splatting (3DGS) [Kerbl et al. 2023] offers an alternative by representing scenes as a mixture of 3D Gaussians, which has shown impressive quality and speed in 3D reconstruction. The real-time rendering of 3DGS has enabled various downstream applications in dynamic scene

reconstruction [G. Wu et al. 2023], editing [Fang et al. 2023; Yiwen Chen et al. 2023] as well as generation [J. Tang et al. 2023].

3D Scene Segmentation. Motivated by the success of radiance fields, several studies have successfully extended such representations to 3D segmentation models. Examples include semantic segmentation [Zhi et al. 2021b; Vora et al. 2021], panoptic segmentation [Bhalgat et al. 2023; Siddiqui et al. 2023b; Fu et al. 2022; Y. Liu et al. 2023] and part segmentation [Zarzar et al. 2022]. A common characteristic of these works is that, given multi-view images of a scene and corresponding 2D labels, these labels are fused into the 3D space as part of the radiance field optimization process. While NeRF has gained significant attention, it is worth noting that the concept of integrating multiple viewpoints and semantic information for 3D reconstruction predates NeRF and has been explored in various methods [Hermans et al. 2014; McCormac et al. 2017; Sünderhauf et al. 2017; L. Ma et al. 2017; Mascaro et al. 2021; Vineet et al. 2015].

More recently, several authors used the Segment Anything model (SAM) [Kirillov et al. 2023] to obtain class-agnostic 3D segmentations [Ying et al. 2024; C. M. Kim et al. 2024a; Cen et al. 2024; Cen et al. 2023; Hu et al. 2024]. SAM offers a promptable approach to segmentation with multiple levels of granularity, which has led most of these studies to adopt an interactive procedure, based on user-provided object seeds. A notable exception is GARField [C. M. Kim et al. 2024a], which operates independently of user input. Instead, it leverages masks produced by SAM, associating each mask with its corresponding 3D scale, and optimizes a scale-conditioned affinity field. GARField can either be interactively queried at specific scales or produce a hierarchy of groupings automatically. However, it does not offer an automatic scale selection mechanism.

3D Feature Distillation. Another related line of work focuses on lifting 2D image features to the 3D space [Tschernezki et al. 2022; Kobayashi et al. 2022a; S. Zhou et al. 2023; Goel et al. 2023]. Similarly to the segmentation case, distilling features from a 2D teacher model, such as DINO [Caron et al. 2021], amounts to optimizing a 3D feature field (alongside the radiance field) to reconstruct the teacher features. In such cases, 3D segmentation involves computing the similarity of the features embedded in the 3D feature field to some query.

Closest to our work are methods that construct 3D language fields from image-text features [Kobayashi et al. 2022a; Kerr et al. 2023; Zuo et al. 2024; Qin et al. 2023; K. Liu et al. 2023; H. Chen et al. 2024; Engelmann et al. 2023], which then enables querying the 3D representation with open-vocabulary text descriptions, thus achieving text-guided segmentation. In particular, LERF [Kerr et al. 2023] is a scaled-conditioned approach that distills multi-scale CLIP [Radford et al. 2021] features into a NeRF model. However, its segmentation quality is limited since it does not use mask supervision and relies mainly on the CLIP encoder which produces global features that lack precise localization. LangSplat [Qin et al. 2023] addresses this issue by combining CLIP features and SAM masks in a 3DGS representation. Unlike LERF or GARfield, it only comprises three distinct scales (the same as SAM), which improves the efficiency of the method but reduces the granularity of the semantic hierarchy.

During inference, both LERF and LangSplat need to densely evaluate the feature field at multiple spatial scales to select the best scale for a given query. This results in a noticeable overhead, especially as the number of scales increases (e.g., as in LERF). Our N2F2 approach makes use of the same underlying 2D models but is simultaneously *more granular* than LangSplat and *more efficient* than both methods while obviating the need for explicit scale selection.

Beyond radiance fields, several works [Peng et al. 2023; Ding et al. 2023; Ha and Song 2022; J. Zhang et al. 2023; Jatavallabhula et al. 2023] operate on point-cloud data and align dense point-features with text and/or image pixels using pre-trained 2D vision-language models.

Hierarchical Representation Learning focuses on developing multi-layered data abstractions to enhance model adaptability and interpretability across tasks. Matryoshka representation learning (MRL) [Kusupati et al. 2022] aims to embed multiple levels of granularity into a single representation allowing the learned embeddings to adapt to the varying computational constraints of the task at hand. MERU [Desai et al. 2023] presents a contrastive approach that yields hyperbolic representations of images and text, resulting in interpretable and structured representations while being competitive with CLIP embeddings on various downstream tasks. In the domain of NLP, TreeFormers [Patel and Flanigan 2022] introduce

a general-purpose text encoder that learns a composition operator and pooling function to construct hierarchical encodings for natural language expressions leading to improvements in compositional generalization. Our approach, inspired by MRL, aims to learn a single feature field wherein different subsets of the feature field encode varying levels of scene properties based on both physical scale and the underlying semantics.

4.3 Method

In this section, we describe **Nested Neural Feature Fields (N2F2)**, an approach to learning multi-scale semantic representations for 3D scenes that uses a form of hierarchical supervision. We first describe the underlying feature field architecture (Section 4.3.1). Then, we introduce the proposed hierarchical supervision method (Section 4.3.2) which encodes varying levels of scene granularity into different subsets of the same feature space, leading to the idea of nested feature fields. Finally, in Section 4.3.3, we formulate a novel composite embedding approach that enables N2F2 to handle compound open-vocabulary queries during inference. Fig. 4.2 provides a high-level overview of N2F2.

4.3.1 Feature Field Architecture

3D Scene Representation.

Given a set of images $\mathcal{I}$ of a scene with a corresponding camera pose $\pi \in SE(3)$ for every image $I \in \mathcal{I}$, we aim to construct a language-aware 3D model of the scene. We use 3D Gaussian Splatting (3DGS) [Kerbl et al. 2023] as the scene representation, where the scene is modeled by a mixture of 3D Gaussians. For any point $x \in \mathbb{R}^3$ in the scene, the influence function of the i^{th} Gaussian is parametrized as:

$$g_i(x) = \exp\left(-\frac{1}{2}(x-\mu_i)^\top \Sigma_i^{-1}(x-\mu_i)\right),$$

where $\mu_i \in \mathbb{R}^3$ is the Gaussian mean or center and $\Sigma_i \in \mathbb{R}^{3\times3}$ is its covariance, expressed by a scaling matrix S_i and rotation matrix R_i as $\Sigma_i = R_i S_i S_i^\top R_i^\top$. To model the 3D radiance field, each Gaussian also has an opacity $\sigma_i \in \mathbb{R}_+$ and a view-dependent colour $c_i(\nu)$ given by spherical harmonic coefficients $\mathcal{C}_i \in \mathbb{R}^k$ (up to the 3^{rd} order). Overall, the 3DGS model is given by $G = \{(\mu_i, \Sigma_i, \sigma_i, \mathcal{C}_i)\}_i$.

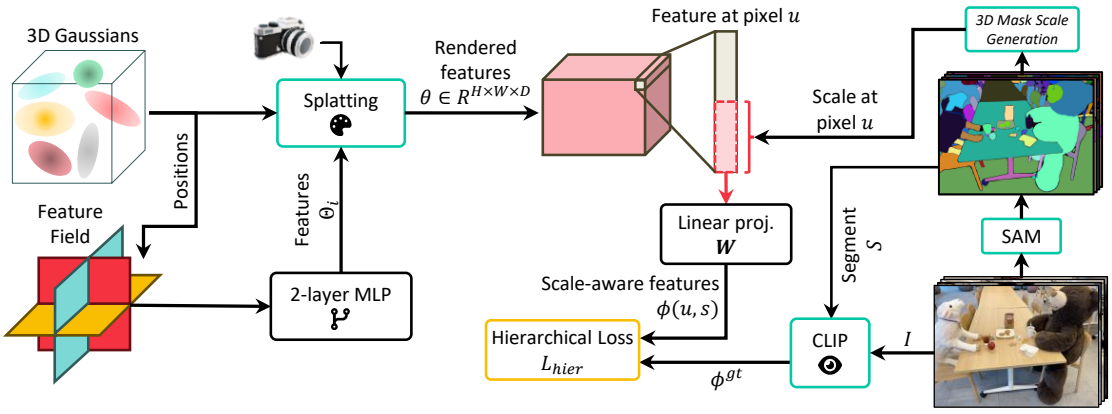


Figure 4.2: **N2F2 Overview.** *Left*: N2F2 employs 3D Gaussian Splatting (3DGS) to represent the scene, augmented with a feature field that captures scene properties across different scales and semantic granularities. *Middle*: Different subsets of the same feature vectors encode scene properties at varying scales. This unified feature field is optimized using a hierarchical supervision loss applied to the scale-aware features. *Right*: We extract a pool of segments using SAM and pre-compute a CLIP embedding for each. Each segment is assigned a physical scale computed using the 3DGS model, which is then used to compute the scale-aware feature.

Given the model G, the 3D radiance field is defined as:

$$\sigma(x) = \sum_i \sigma_i g_i(x), \quad c(x,\nu) = \frac{\sum_i c_i(\nu)\sigma_i g_i(x)}{\sum_j \sigma_j g_j(x)}. \tag{4.1}$$

Given G and a camera viewpoint π, the differentiable Gaussian Splatting renderer [Kerbl et al. 2023] produces an image $\hat{I} = \mathcal{R}(G,\pi) \in \mathbb{R}^{H\times W\times 3}$. The key reason for choosing 3DGS as the scene representation rather than NeRF [Mildenhall et al. 2020] or its faster variants [Sara Fridovich-Keil and Alex Yu et al. 2022; Müller et al. 2022; A. Chen et al. 2022; C. Sun et al. 2022] is that 3DGS provides a very efficient renderer (both in speed and memory) resulting in real-time rendering of full-sized images as well as feature maps, which allows us to perform rapid querying at inference time.

Gaussian Feature Field.

In this work, we propose to augment the 3DGS model with a feature field $\Theta : \mathbb{R}^3 \to \mathbb{R}^D$. A natural choice for doing this would be to associate each Gaussian with a learnable feature vector and optimize these features in the same manner as the other parameters (i.e. $\mu_i, \Sigma_i, \sigma_i, \mathcal{C}_i$). However, in practice, the size D tends to very high (e.g., $D = 512$ for CLIP embeddings) which drastically increases the number of parameters in the augmented 3DGS model. To address this issue, we model

the feature field Θ with a memory-efficient TriPlane representation [Fridovich-Keil et al. 2023] followed by a 3-layer MLP. During rendering, we query this hybrid representation at the Gaussian centers to get the associated feature for each Gaussian, i.e. $\Theta_i = \Theta(\mu_i)$.[1] Thus, given a camera viewpoint π, this results in a rendered pixel-level feature map:

$$\theta = \mathcal{R}(\Theta, G, \pi) \in \mathbb{R}^{H \times W \times D} \tag{4.2}$$

In addition, to save memory during training, we first render the TriPlane features, then use them as input to the 3-layer MLP to obtain the pixel/ray feature. This deferred rendering is performed only during training and leads to a significantly faster optimization process without any loss in performance. Please refer to implementation details in the appendix for differences to test time.

4.3.2 Scale-aware Hierarchical Supervision

Our main goal is to learn a unified representation that captures the meaning of the scene across different scales and semantic granularities. We do so by modeling N2F2 with a *single* Gaussian feature field, where different feature dimensions represent varying levels of detail, from the overall scene structure to fine-grained object particulars. To achieve this, we train our feature field model using a scale-aware hierarchical supervision method described below.

Extracting Training Data.

To get an accurate geometry model for each scene, we first optimize the radiance field related parameters of the 3DGS model. Then, we use SAM [Kirillov et al. 2023] to extract class-agnostic but semantically meaningful segments from every image $I \in \mathcal{I}$, resulting in a pool of segments $\mathcal{S}$ spanning all the images. Following GARField [C. M. Kim et al. 2024a], for each segment $S \in \mathcal{S}$, we use the expected depth values from the 3DGS model to lift the pixels in the segment to a corresponding 3D point cloud. We obtain the scale of this point cloud as the largest eigenvalue of the covariance matrix of the 3D point positions. We then quantize the extracted scales into D bins using a quantile transformation, where

[1] We use the same variable name Θ_i for simplicity.

D is the dimension of the feature vectors in our field.[2] We denote the quantized scale corresponding to segment S as $s \in [0, 1)$. Next, for every segment $S \in \mathcal{S}$ of an image I, we use the CLIP [Radford et al. 2021] image encoder to obtain a language embedding $\phi^{\text{gt}} \in \mathbb{R}^D$:

$$\phi^{\text{gt}} = E(I \odot S), \tag{4.3}$$

where E is the CLIP vision encoder and $\odot$ is the Hadamard product.

We use this vector as "ground truth" to optimize our N2F2 model with hierarchical supervision. To do this, we associate each dimension of the feature field $\Theta \in \mathbb{R}^D$ to a quantized scale value (note that the dimension D is also the number of quantized values), such that the lower dimensions are mapped to larger (coarser) scales and higher dimensions to finer scales. This is predicated on the observation that objects or scene components at larger scales can be differentiated with fewer dimensions, using less specific features. Conversely, as one zooms into finer scales, more dimensions become necessary to match the increased specificity and diversity of the features. Mathematically, for a quantized scale $s \in [0, 1)$, we associate the dimension $M(s)$ defined by the mapping $M(s) = \lceil D \cdot (1 - s) \rceil$, where $\lceil \cdot \rceil$ is the ceiling function.

Scale-aware Feature.

Consider a pixel u sampled from a segment S with scale s. The rendered feature $\theta(u)$ can be obtained from our feature field using Eq. 4.2. Now, given the mapping $M(\cdot)$, we obtain the *scale-aware* feature as:

$$\phi(u, s) = \mathbf{W}_{1:M(s)} \theta(u)_{1:M(s)} \tag{4.4}$$

Here, $\mathbf{W} \in \mathbb{R}^{D \times D}$ is a learnable projection matrix. $\mathbf{W}_{1:M(s)} \in \mathbb{R}^{D \times M(s)}$ is formed by the first $M(s)$ columns of $\mathbf{W}$ and $\theta(u)_{1:M(s)}$ are the first $M(s)$ elements of the feature $\theta(u)$. Hence, the resulting scale-aware feature $\phi(u, s) \in \mathbb{R}^D$. We do not explicitly normalize ϕ or θ so that the former remains a linear function of the latter.

[2] This transforms the scales to follow a uniform distribution and also reduces the impact of marginal outliers.

Intuitively, the projection matrix $\mathbf{W}_{1:M(s)}$ dynamically adjusts to the variable dimensionality of the feature vector $\theta(u)_{1:M(s)}$ corresponding to pixel u at scale s. It ensures that this vector is mapped to a fixed dimensionality D that matches that of the features ϕ^{gt}. This enables the use of *single-scale* features to supervise the training of a nested *multi-scale* feature field.

Hierarchical Loss Supervision.

During training, we sample pixels u uniformly across the image set $\mathcal{I}$. Since there can be multiple segments associated with the same pixel, we sample a segment S with a probability inversely proportional to the log of the area of S. With the rendered features $\theta(u)$ and the language-aligned teacher embeddings ϕ^{gt} from Eq. 4.3, we minimize the loss:

$$\mathcal{L}_{\text{hier}} = \sum_{u,s} \mathcal{L}(\phi(u,s), \phi^{\text{gt}}) \tag{4.5}$$

where $\mathcal{L} = \mathcal{L}_2 + \lambda \mathcal{L}_{\cos}$ is a weighted combination of the L2 and cosine distances.

4.3.3 Composite Embedding for Open-Vocabulary Querying

Once the model has been optimized with the hierarchical loss function, it can be queried with language prompts encoded using the CLIP text encoder. The querying can be done either in 3D with the point features at the Gaussian centers or in 2D with the rendered 2D features. We discuss the latter in this section, but our method is compatible with both scenarios.

Previous methods [Kerr et al. 2023; Qin et al. 2023], compute relevancy maps at multiple scales (e.g., LERF uses 30 scales) for each text query and then choose the scale with the highest relevancy score. We notice that this leads to a noticeable overhead for every text input, especially for a large number of scales. Hence, we propose a *composite embedding* method that allows us to compute only *one* relevancy map per text query and use that as the output.

We define the *composite embedding* at a pixel u to be a weighted sum of all possible scale-aware features at that pixel. The idea of using a linear combination of features across scales arises from the necessity to handle compound queries effec-

tively. Consider the query "lid of a cup". The scale-aware features ideally trigger twice for pixels that are both on a "lid" and on a "cup", akin to an intersection of sets. This is because the composite embedding aggregates features from the scales corresponding to "a cup" and "a lid", effectively highlighting the summed response of these features. Conversely, querying "a cup" would trigger once for all pixels on a cup, reflecting the broader, single-scale interpretation.

The weights in the linear combination are *query-agnostic* and account for the relevance of each scale for *any* given query. Their goal is to select scales that are more strongly related to specific types of queries, and are obtained by comparing each scale-specific feature vector to prototypical phrases such as "*object*", "*stuff*", "*thing*", "*part*" and "*texture*".[3] Mathematically, given a point feature for a Gaussian, Θ_i, the weight γ_i^{3D} is:

$$\gamma_i^{\text{3D}} = \underset{d}{\text{Softmax}}\left(\max_k \left(\mathbf{W}_{1:d}\Theta_{i,1:d}\right)^\top \phi_k^{\text{canon}}\right), \tag{4.6}$$

where $\{\phi_k^{\text{canon}}\}$ is the set of CLIP embeddings of the predefined canonical phrases listed above. The γ^{3D} tensor only needs to be computed once post-training. For a new viewpoint, γ^{3D} is rendered to obtain the pixel-level tensor $\gamma \in \mathbb{R}^{H\times W\times D}$.

Since there are exactly D quantized scales, each mapped to a dimension of the feature field, the composite embedding is $\phi_{\text{comp}}(u) = \sum_{d=1}^{D} \gamma_d(u)\phi(u, s_d)$. Expanding this using Eq. 4.4, we get:

$$\begin{aligned}\phi_{\text{comp}}(u) &= \sum_{d=1}^{D} \gamma_d(u)\mathbf{W}_{1:d}\theta(u)_{1:d} = \sum_{d=1}^{D} \gamma_d(u)\left(\sum_{j=1}^{d}\theta(u)_j W_j\right)\\ &= \sum_{j=1}^{D}\left(\sum_{d=j}^{D}\gamma_d(u)\right)\theta(u)_j W_j\end{aligned}$$

We can rewrite this as $\phi_{\text{comp}}(u) = \mathbf{W}\tilde{\theta}(u)$, where

$$\tilde{\theta}(u) = \begin{bmatrix}\sum_{d=1}^{D}\gamma_d(u)\\ \sum_{d=2}^{D}\gamma_d(u)\\ \vdots\\ \gamma_D(u)\end{bmatrix} \odot \theta(u), \quad \text{or equivalently,} \quad \tilde{\Theta}_i = \begin{bmatrix}\sum_{d=1}^{D}\gamma_{i,d}^{\text{3D}}\\ \sum_{d=2}^{D}\gamma_{i,d}^{\text{3D}}\\ \vdots\\ \gamma_{i,D}^{\text{3D}}\end{bmatrix} \odot \Theta_i \tag{4.7}$$

[3] LERF [Kerr et al. 2023] uses similar phrases to compute relevancy scores *per query*.

Note, $\gamma_{i,d}^{3D}$ denotes the d^{th} element of the vector γ_i^{3D} corresponding to the i^{th} Gaussian. Given the above formulation, instead of computing the point-features Θ_i, we pre-compute the point-features $\tilde{\Theta}_i$ prior to the text-querying process.

4.4 Experiments

Tasks and Datasets.

We assess the ability of N2F2 to perform open-vocabulary *segmentation* and *localization* on challenging natural language queries. We train and evaluate our method on the expanded LERF dataset made available by Qin *et al.* [Qin et al. 2023]. This expanded version contains ground-truth segmentation masks for textual queries required to evaluate the segmentation performance. The expanded version also contains additional and more challenging localization samples to evaluate the localization accuracy. We also use the 3D-OVS dataset [K. Liu et al. 2023] to evaluate and compare our open-vocabulary 3D segmentation capabilities.

Metrics.

For the 3D localization task, we consider a text query to be successfully localized if the identified highest-relevancy pixel in a view lies inside the ground-truth bounding box, reporting the localization accuracy. We report the mIoU scores for the 3D segmentation task.

Baselines.

We compare our method with LERF [Kerr et al. 2023] and LangSplat [Qin et al. 2023], which are the current state-of-the-art feature distillation methods for open-vocabulary 3D segmentation and localization. We also compare with a baseline distilling the pixel-aligned feature from LSeg [B. Li et al. 2022] into 3D. Note that, both LangSplat and our method utilize SAM [Kirillov et al. 2023] during optimization.

Table 4.1: Localization accuracy (%) comparisons on LERF scenes, averaged across text queries. More query-specific results are provided in the appendix. The second column indicates if a method utilizes segmentation from SAM [Kirillov et al. 2023] during training.

Method	SAM	*bouquet*	*ramen*	*figurines*	*teatime*	*waldo_kitchen*	Overall
LSeg [B. Li et al. 2022]		50.0	14.1	8.9	33.9	27.3	21.1
LERF [Kerr et al. 2023]		**91.7**	62.0	75.0	84.8	72.7	73.6
LangSplat [Qin et al. 2023]	✓	—	73.2	80.4	88.1	**95.5**	84.3
N2F2 (**Ours**)	✓	**91.7**	**78.8**	**85.7**	**91.5**	**95.5**	**88.6**

Table 4.2: Open-vocabulary 3D semantic segmentation performance (mIoU) on the expanded LERF dataset from Qin *et al.* [Qin et al. 2023], averaged across queries. The second column indicates if a method utilizes segmentation from SAM [Kirillov et al. 2023] during training.

Method	SAM	*ramen*	*figurines*	*teatime*	*waldo_kitchen*	Overall
LSeg [B. Li et al. 2022]		7.0	7.6	21.7	29.9	16.6
LERF [Kerr et al. 2023]		28.2	38.6	45.0	37.9	37.4
LangSplat [Qin et al. 2023]	✓	51.2	44.7	65.1	44.5	51.4
N2F2 (**Ours**)	✓	**56.6**	**47.0**	**69.2**	**47.9**	**54.4**

4.4.1 Results

3D Localization.

Table 4.1 shows the 3D localization performance on the expanded LERF dataset [Qin et al. 2023], on a diverse set of scenes, namely *bouquet*, *ramen*, *figurines*, *teatime*, and *waldo_kitchen*. Notably, the *bouquet* scene is not a part of the extended LERF dataset, and LangSplat [Qin et al. 2023] does not provide results for this scene. Thus, for this scene, we evaluate and compare on the text queries from the original LERF dataset. Our method N2F2 significantly outperforms the existing state-of-the-art methods, LERF [Kerr et al. 2023] and LangSplat [Qin et al. 2023], across most of the evaluated scenes, achieving an overall accuracy of 88.6%. Our N2F2 approach particularly outshines prior work in samples with complex *compound queries*, such as "sake cup", "bag of cookies", etc. For a detailed breakdown of the performance across such queries, please refer to Section 4.7.1.

3D Segmentation.

Tab. 4.2 and4.3 demonstrate the performance on the open-vocabulary 3D segmentation task on the expanded LERF [Qin et al. 2023] and 3D-OVS [K. Liu et al.

Figure 4.3: Qualitative comparisons with LangSplat [Qin et al. 2023] on challenging *compound* queries.

2023] datasets respectively. The expanded LERF dataset contains various *compound queries*, such as "bag of cookies", "coffee mug", and "paper napkin", in contrast to simple queries like "cookies", "mug", "apple", which makes it challenging to segment these referred objects. LERF and LangSplat, which directly distill CLIP embeddings, struggle with such queries, producing a high relevancy score for all objects/attributes in the compound query. An example shown in Fig. 4.3 shows that for "bag of cookies", LERF highlights both the "bag" *and* the "cookies", although our method correctly highlights the "bag" only. More such examples are included in the appendix. Overall, our method achieves an mIoU of 54.5 and outperforms both LERF and LangSplat on all scenes by a significant margin.

The 3D-OVS dataset contains relatively simple scenes where all included objects are of similar scales. Thus, the performance of previous state-of-the-art methods is highly saturated with LangSplat achieving almost perfect segmentations for many text queries. Nevertheless, our method outperforms LangSplat on 4 out of 6 scenes and improves the overall mIoU by 0.5%. More qualitative results are included in the appendix.

Table 4.3: Open-vocabulary 3D semantic segmentation performance (mIoU) on the 3D-OVS dataset [K. Liu et al. 2023], averaged across queries. The second column indicates if a method utilizes segmentation from SAM [Kirillov et al. 2023] during training.

Method	SAM	*bed*	*bench*	*room*	*sofa*	*lawn*	Overall
LSeg [B. Li et al. 2022]		56.0	6.0	19.2	4.5	17.5	20.6
ODISE [J. Xu et al. 2023]		52.6	24.1	52.5	48.3	39.8	43.5
OV-Seg [Liang et al. 2023]		79.8	88.9	71.4	66.1	81.2	77.5
FFD [Kobayashi et al. 2022b]		56.6	6.1	25.1	3.7	42.9	26.9
LERF [Kerr et al. 2023]		73.5	53.2	46.6	27	73.7	54.8
3D-OVS [K. Liu et al. 2023]		89.5	89.3	92.8	74	88.2	86.8
LangSplat [Qin et al. 2023]	✓	92.5	**94.2**	**94.1**	90.0	96.1	93.4
N2F2 (**Ours**)	✓	**93.8**	92.6	93.5	**92.1**	**96.3**	**93.9**

4.4.2 Ablation Studies

Composite Embedding vs Explicit Scale Selection.

Unlike LERF or LangSplat, our method does not explicitly perform scale selection for every query. Instead, we design a composite embedding (and the γ tensor, *c.f.* Eq. 4.7) to produce high relevancy at the appropriate scales and generate an aggregated relevancy map *implicitly*. But we could, in principle, also explicitly search the scale with the highest relevancy score in the same manner as LERF/LangSplat. Table 4.4 shows a comparison between the composite embedding and explicit scale selection for localization and segmentation on the expanded LERF dataset. We observe that the composite embedding performs better than the explicit scale selection method while being approximately 5× faster during querying. Compared to Table **??**, we note that the N2F2 explicit scale selection baseline *also* outperforms LangSplat [Qin et al. 2023], which also uses explicit selection over 3 scales. This further demonstrates the advantage of our hierarchical representation with increased granularity (i.e. D scales).

Additionally, we ask — *what performance can N2F2 achieve if the ideal scale was somehow provided?* We call this the "Oracle" scale. For a given text query, we define the oracle scale to be the scale that gives the highest performance for the task in question. This gives an upper bound on the performance achievable by our method, which is shown in Table 4.4.

Table 4.4: Comparison of composite embedding based querying vs explicit scale-selection based querying on 4 LERF scenes. **Top**: 3D localization. **Bottom**: 3D segmentation.

Method	*ramen*	*figurines*	*teatime*	*waldo_kitchen*	Overall
3D Localization					
Composite embedding	**78.8**	**85.7**	**91.5**	**95.5**	**87.9**
Explicit scale selection	**78.8**	83.9	89.8	**95.5**	87.0
Oracle scale (Upper bound)	83.0	91.0	93.2	95.5	90.6
3D Segmentation					
Composite embedding	**56.6**	**47.0**	**69.2**	**47.9**	**54.4**
Explicit scale selection	55.7	45.9	66.8	46.3	53.7
Oracle scale (Upper bound)	59.8	49.1	71.2	49.2	57.3

Table 4.5: Breakdown of inference-time querying speed for different methods.

Method	Rendering time	Time (per query)	Time (10 queries)	Total time (1 query)	Total time (10 queries)
LERF	18.4s	2.5s	20.0s	20.9s	38.4s
LangSplat	0.05s	0.29s	2.5s	0.3s	2.55s
N2F2 (**Ours**)	0.1s	0.16s	1.4s	**0.24s**	**1.5s**

Efficiency.

For each text query, LERF computes relevancy maps at 30 scales (from 0 to 2) and LangSplat computes them at 3 scales hardcoded to the SAM [Kirillov et al. 2023] hierarchy. For N2F2, with the composite embeddings, we can get the final output by computing only a single relevancy map per text query. This leads to a significant speed up during querying making our method overall 1.7× faster than LangSplat and 26× faster than LERF, while using more scales.

Table 4.5 shows a full breakdown of the querying time cost for each method. With our method, we pre-compute γ^{3D} for every Gaussian in 3D. Then, for each query, we render the 512-dimensional composite embedding, in chunks. Our method takes 0.1s to produce a 730×987×512 feature map from one viewpoint, while LangSplat takes 0.05s to render and decode features to the same size. Once rendered, LangSplat takes about 0.25s/query, while our method takes about 0.16s/query resulting in a 1.7x speedup amortized over 10 queries per viewpoint.

Table 4.6: Effect of step-size in the dimension-scale mapping. Evaluation on the LERF *teatime* scene for the open-vocabulary 3D segmentation task (mIoU).

Scene	$k = 1$	$k = 8$	$k = 32$	$k = 128$	$k = 170$
LERF (*teatime*)	**69.2**	68.7	68.1	67.2	66.4

Effect of step-size in the dimension-scale mapping.

In Section 4.3.2, we quantized the object scales to D values, thereby associating each of the dimensions from $\{1, 2, \ldots, D\}$ with a quantized scale in the hierarchical loss optimization (*c.f.* Eq. 4.5). We investigate what happens when we do not utilize all the dimensions for scale supervision. That is, with a "*step-size*" of k in the dimension-scale mapping, we quantize the scales to $\lfloor D/k \rfloor$ values and associate them to the dimensions $\{k, 2k, \ldots, k * \lfloor D/k \rfloor\}$. For example, with $k = 3$, only the dimensions $\{3, 6, \ldots, 510\}$ are used. We train models with $k = \{8, 32, 128, 170\}$ using the hierarchical loss and compare them with the default model ($k = 1$) on the LERF *teatime* scene for open-vocabulary 3D segmentation. Table 4.6 shows that performance gradually decreases as the granularity of the feature field is reduced, i.e. when we reduce the utilization of its dimensions. We note that the case when $k = 170$, i.e. when only 3 different quantized scales or dimensions are used (namely $\{170, 340, 510\}$), is the closest to LangSplat which utilizes 3 scales. In this setting, we achieve an mIoU of 66.4 which is 1.3 points higher than LangSplat's performance (*c.f.* Table 4.2).

4.5 Limitations

Despite the advancements N2F2 brings to hierarchical scene understanding, it faces certain challenges. The model's ability to learn accurate feature fields heavily depends on the quality of scene reconstructions, necessitating a diverse array of images from varied viewpoints. While N2F2 accurately handles compound noun phrases (like "coffee mug") and partitive constructions ("bag of cookies" or "lid of the cup"), it struggles with global scene context queries, such as identifying a "wooden desk in the corner of the room". This limitation points to the challenge of integrating broad scene comprehension with specific object identification. Addressing these will be crucial for future developments in the field.

4.6 Conclusion

In this work, we introduced Nested Neural Feature Fields (N2F2), a novel approach for hierarchical scene understanding. N2F2 employs scale-aware hierarchical supervision to encode scene properties across multiple granularities within a unified feature field, significantly advancing open-vocabulary 3D segmentation and localization tasks. Through extensive experiments, our approach demonstrated superior performance over state-of-the-art methods, such as LERF [Kerr et al. 2023] and LangSplat [Qin et al. 2023], highlighting the effectiveness of our hierarchical supervision methodology. In particular, our method outshines on complex compounded and partitive constructions such as "bag of cookies", "chair legs", "blueberry donuts", etc. Finally, we also propose a novel *composite embedding* design that enables highly efficient querying as well as better performance than explicit scale-selection methods used in previous works.

Ethics. For further details on ethics, data protection, and copyright please see https://www.robots.ox.ac.uk/~vedaldi/research/union/ethics.html.

Acknowledgements

We are grateful for funding from EPSRC AIMS CDT EP/S024050/1 and AWS (Y. Bhalgat), ERC-CoG UNION 101001212 (A. Vedaldi and I. Laina), EPSRC VisualAI EP/T028572/1 (I. Laina, A. Vedaldi and A. Zisserman), and Royal Academy of Engineering RF\201819\18\163 (J. Henriques).

4.7 Appendix

4.7.1 Performance on Compound Queries

In Sec. 4.1 of the paper, we noted that, in comparison to previous methods (LERF [Kerr et al. 2023] and LangSplat [Qin et al. 2023]), N2F2 offers a strong advantage on queries which are *compound* or *partitive* in nature. To understand this effect more clearly, we split the text queries from the expanded LERF dataset into two sets: (1) *simple* single word queries such as "cookies", "cup", "waldo",

Table 4.7: Breakdown of performance on *simple* and *compound* queries for open-vocabulary 3D segmentation (mIoU). "comp." is short for compound queries.

Method	*ramen*		*figurines*		*teatime*		*waldo_kitchen*	
	simple	comp.	simple	comp.	simple	comp.	simple	comp.
LangSplat [Qin et al. 2023]	65.4	35.1	53.9	32.7	70.1	47.3	52.6	31.3
N2F2 (**Ours**)	**66.2**	**42.3**	**54.7**	**36.1**	**72.2**	**50.9**	**53.2**	**35.5**

"spatula", (2) *compound or partitive* queries[4] such as "bag of cookies", "frog cup", "dark cup", "toy elephant", "toy cat statue". Table 4.11 provides a full list of compound queries for each scene.

Table 4.7 shows that N2F2 significantly outperforms LangSplat [Qin et al. 2023], the current state-of-the-art method, especially on compound queries where we see an improvement as big as **7.2** mIoU points on the *ramen* scene.

To gain a clearer insight into which queries N2F2 excels at, we rank the queries within each scene according to the performance disparity between N2F2 and LangSplat. We then highlight the top-3 and bottom-3 queries that exhibit the largest and smallest differences in performance, respectively. As shown in Table 4.8, the largest performance improvements (*c.f.* the 'top-3 queries' column) are observed on compound queries and especially the ones which could apply to multiple objects, if they were not specific. For example, the *figurines* scene contains two similar looking rubber ducks, but only one of them wears a hat. Hence, the query "rubber duck with hat" must only segment the referred duck. This is successfully achieved by N2F2 but LangSplat segments parts of both the ducks resulting in a lower segmentation IoU.

Fig. 4.5, 4.6, and 4.7 demonstrate more qualitative comparisons between N2F2 and LangSplat [Qin et al. 2023] on various challenging *compound* queries.

4.7.2 Performance with a weaker segmenter

Following LangSplat [Qin et al. 2023], we use the Segment Anything model (SAM) [Kirillov et al. 2023] to extract class-agnostic segments for every image and use these segments to compute CLIP [Radford et al. 2021] embeddings for supervision (*c.f.* Eq. 4.3). To understand the effect of the choice of 2D segmenter on the final

[4]For brevity, we refer to *all* such queries as compound queries.

Table 4.8: Queries within each scene ranked according to performance disparity (Δ_{perf}) between N2F2 and LangSplat [Qin et al. 2023]. Top-3 and bottom-3 ranked queries are shown.

Scene	Top-3 queries	Δ_{perf}	Bottom-3 queries	Δ_{perf}
ramen	sake cup	+9.8	chopsticks	+0.0
	spoon handle	+6.9	bowl	+0.5
	wavy noodles	+5.1	egg	+0.5
figurines	rubber duck with hat	+18.1	pumpkin	−0.1
	red apple	+8.6	waldo	+0.3
	porcelain hand	+4.5	pikachu	+0.4
teatime	bag of cookies	+11.7	plate	−0.3
	stuffed bear	+7.0	sheep	+0.1
	bear nose	+5.4	dall-e brand	+0.5
waldo_kitchen	Stainless steel pots	+10.7	knife	−0.5
	pour-over vessel	+7.2	ottolenghi	+0.2
	red cup	+6.0	pot	+0.3

Table 4.9: Performance comparison between our method and LangSplat [Qin et al. 2023] using Detic [X. Zhou et al. 2022] as 2D segmenter.

Method	*figurines*	*teatime*
LangSplat w/ Detic	43.2	63.9
LangSplat w/ SAM (*original*)	44.7	65.1
N2F2 w/ Detic	**45.5**	**66.2**
N2F2 w/ SAM (*original*)	**47.0**	**69.2**

performance, we optimize our method as well as LangSplat using segments from a weaker model, namely Detic [X. Zhou et al. 2022]. Table 4.9 shows that N2F2 (with Detic) performs slightly worse than N2F2 (with SAM), but still outperforms LangSplat (with SAM).

4.7.3 Experiments with scenes from ScanNet dataset

In this work, we mainly evaluate on the LERF and 3D-OVS datasets. To expand this evaluation, we also considering evaluating and comparing our method on the ScanNet dataset [Dai et al. 2017]. Table 4.10 shows the performance comparison on 4 scenes: *0050_02*, *0144_01*, *0300_01* and *0423_02*.

Table 4.10: Performance on ScanNet dataset.

Method	*0050_02*	*0144_01*	*0300_01*	*0423_02*
LangSplat	60.2	56.5	54.7	57.0
N2F2	**68.1**	**61.7**	**59.9**	**60.4**

4.7.4 Analysis of backbone components

We analyze the different components of the N2F2 architecture and evaluate their impact on the final performance. Specifically, we compare using a pure-MLP backbone instead of the TriPlane+MLP backbone used in N2F2. We empirically observe that the MLP backbone is 2.3× more memory-efficient than TriPlane+MLP. Notably, both these backbones use 7× and 3× less memory than LangSplat [Qin et al. 2023] respectively. Both backbones achieve similar final performance, but TriPlane+MLP converges 20× faster than the MLP backbone.

Additionally, we also experiment with removing the nested feature design in N2F2. We observe that this results in a drop of 2.0 mIoU on segmentation and 2.5% on localization, still outperforming LangSplat.

4.7.5 Open-vocabulary Retrieval Task

Next, we assess the performance of our method in the context of text-based retrieval. The objective of the open-vocabulary retrieval task is to identify the most relevant object within the scene based on a natural language query. Unlike segmentation, which assesses the precision of object boundaries, retrieval focuses more on the model's ability to find relevant objects.

To this end, we repurpose the expanded LERF dataset from Qin *et al.* [Qin et al. 2023] for this task. For a test view, we take the pool of segments from SAM as given, compute a relevancy score *per segment* (which is the average relevancy over segment pixels), and then rank these segments according to the predicted relevancy scores. In this task, for a given text query, if the correct corresponding segment is within the top K retrieved segments, then the query is said to be retrieved. We report the recall ($R@K$) metric which is the fraction of correctly retrieved queries for a given K. Note that the performance at $K = 1$ is correlated with the localization accuracy of the method, indicating its ability to directly pinpoint the exact segment most relevant to the given text query. Fig. 4.4 compares the

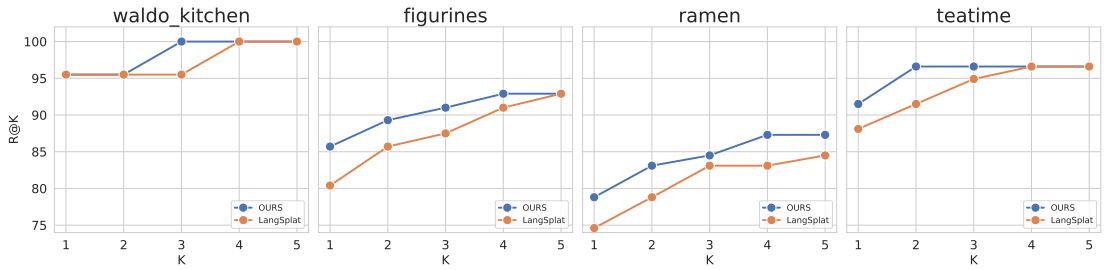


Figure 4.4: Open-vocabulary Retrieval performance on the expanded LERF dataset from Qin *et al.* [Qin et al. 2023]. $R@K$ (%) reported at $K = 1, 2, 3, 4, 5$ for each scene.

performance of our method with LangSplat [Qin et al. 2023] across a range of K. We can see that at $K = 1$, N2F2 consistently performs better than LangSplat across scenes (which was also reflected in the localization performance reported in Table 1 of the main paper). The $R@K$ of both methods increases with K, while the performance gap becomes smaller.

4.7.6 Implementation Details

Extracting Training data

To obtain embeddings for the text queries, we leverage the OpenCLIP ViT-B/16 model with the segment masks obtained from the SAM ViT-H model. We use the `laion2b_s34b_b88k` pretrained checkpoint for the OpenCLIP model. For SAM mask generation, we sample pixels on a 32×32 grid, set the minimum mask region area to be 100 and use a IoU threshold of 0.7. To get an accurate geometry model for each scene, we first optimize the radiance field related parameters of the 3D Gaussian Splatting (3DGS) model using the RGB images for 30k iterations. We use the expected depth obtained from the 3DGS model to lift the SAM masks into 3D and obtain a physical scale for each mask as the largest eigenvalue of the covariance matrix of the lifted 3D segment-pixels.

Feature Field Optimization

Then, we optimize our feature field for another 30k iterations while freezing all other parameters of the model. As mentioned in the paper, the feature field is modeled with a TriPlane + 3-layer MLP. We use a 512×512 resolution for each of the planes with 64-dimensional features. A hidden size of 256 is used for the MLP. We empirically observe that using a *zero*-initialization for the projection matrix

$\mathbf{W}$ gives the best results. The λ weight used with the cosine loss (*c.f.* Eq. (5)) is set to be 0.001. We use a learning rate of 0.0016 × scene_extent for the TriPlane and 0.00125 for the MLP. All experiments and comparisons used an *NVIDIA P40* (24GB RAM). Our models are trained in ≈1 hour and take ≈600MB of memory.

Rendering details

As described in Sec. 3.1 of the paper, during rendering, we query the TriPlane+MLP representation at the Gaussian centers to get the associated feature for each Gaussian, and then use the Gaussian Splatting renderer to obtain the rendered feature map. The default 3DGS implementation does not support rendering arbitrary features, so we modify and use the *NDRasterizer* provided by Nerfstudio [Tancik et al. 2023] to render the features.

Deferred rendering during training and not during testing

As described in Sec. 3.1 of the paper, we use deferred rendering to save memory during training, i.e. e first render TriPlane features and then apply the MLP to obtain the pixel/ray feature. We do not do this during test time. This is because, as described in Sec. 3.3, we premultiply point features with the γ^{3D} tensor to obtain the composite embedding. Hence, we need to maintain full-sized features per Gaussian during test-time.

Note that, applying the MLP on the rendered features (i.e. $\text{MLP}(\alpha_1 F_1 + \alpha_2 F_2 + \dots)$) is not strictly equal to first applying the MLP on the point features and then rendering the full features (i.e. $\alpha_1 \text{MLP}(F_1) + \alpha_2 \text{MLP}(F_2) + \dots$). However, after training, Gaussians tend to be either transparent or opaque, so most pixels receive a non-negligible contribution from a single dominant Gaussian ($\alpha_i \approx 1$), making the train-time and test-time formulations approximately equivalent.

Efficient Scale-aware Feature Computation

When optimizing the Hierarchical Loss (*c.f.* Eq. (5)), different pixels u in a batch will have different associated scales s (and hence different mapped dimensions $M(s)$). Thus, to efficiently compute the scale-aware features $\mathbf{W}_{1:M(s)}\theta(u)_{1:M(s)}$, we implement a batch-wise masking mechanism. Specifically, for a given scale s, we employ a binary mask $B(s)$ where entries corresponding to the active dimensions

Table 4.11: Distinction between the *simple* and *compound* queries for each scene.

Scene	Simple queries	compound queries
waldo_kitchen	sink, refrigerator, cabinet, spatula, toaster, plate, ottolenghi, spoon, ketchup, pot, knife	yellow desk, Stainless steel pots, frog cup, red cup, pour-over vessel, plastic ladle, dark cup
figurines	jake, bag, spatula, porcelain hand, rubics cube, waldo, pumpkin, miffy, pirate hat, old camera, pikachu	tesla door handle, rubber duck with hat, red toy chair, toy elephant, green toy chair, pink ice cream, green apple, red apple, rubber duck buoy, toy cat statue
ramen	corn, plate, chopsticks, egg, bowl, kamaboko, onion segments, napkin, spoon, hand, nori	sake cup, glass of water, wavy noodles
teatime	three cookies, plate, hooves, apple, dall-e brand, coffee, sheep	bear nose, bag of cookies, tea in a glass, stuffed bear, coffee mug, paper napkin, yellow pouf

$1 : M(s)$ are set to 1, and all others are set to 0. The scale-aware features for each pixel u are then computed as $\mathbf{W}_{1:M(s)}\theta(u)_{1:M(s)} = \mathbf{W} \cdot (B(s) \odot \theta(u))$, where $\odot$ denotes element-wise multiplication. The mask B is computed batch-wise with negligible overhead.

4.7.7 Qualitative results

In Fig. 4.5, 4.6, and 4.7, we show qualitative comparisons between N2F2 and LangSplat [Qin et al. 2023] on various challenging *compounded* queries.

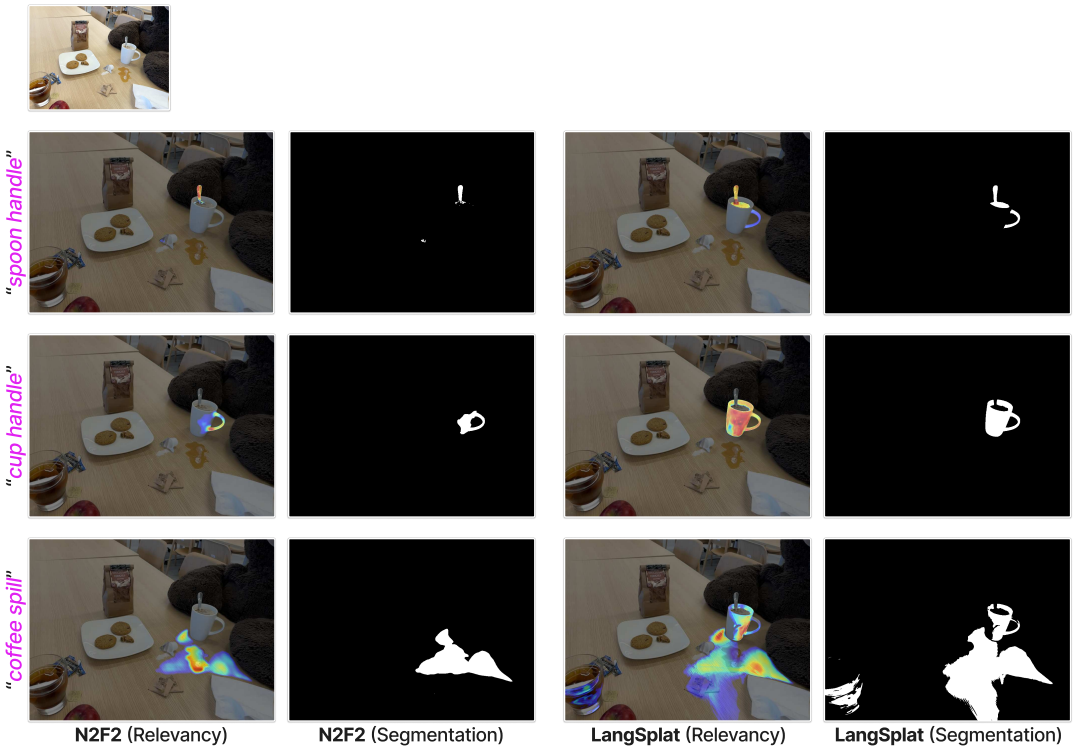


Figure 4.5: Scene: *teatime*. Each row contains results for the text query shown on the **left**. **Columns**: N2F2 (Relevancy and Segmentation maps), LangSplat [Qin et al. 2023] (Relevancy and Segmentation maps).

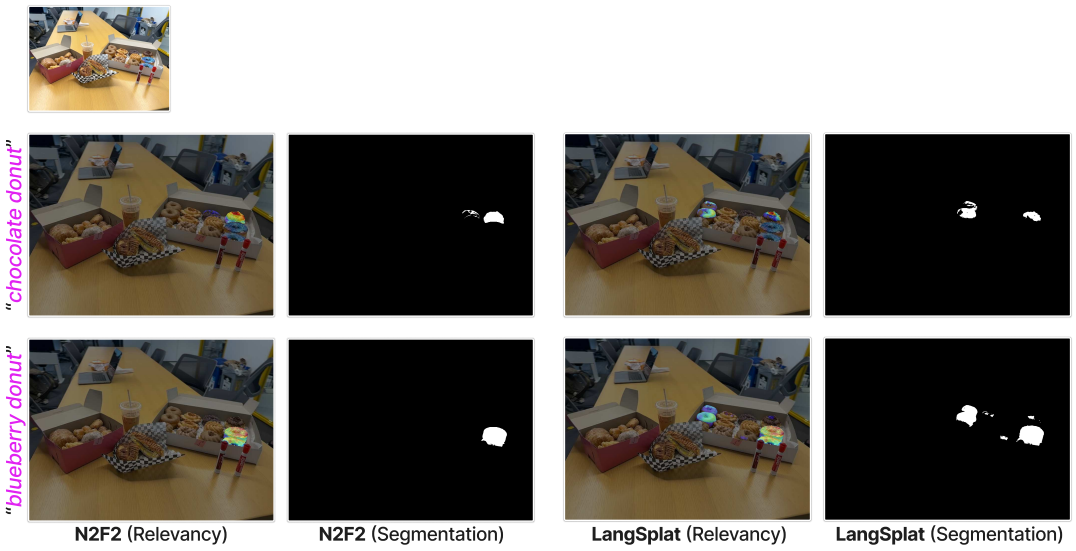


Figure 4.6: Scene: *donuts*. Each row contains results for the text query shown on the **left**. **Columns**: N2F2 (Relevancy and Segmentation maps), LangSplat [Qin et al. 2023] (Relevancy and Segmentation maps).

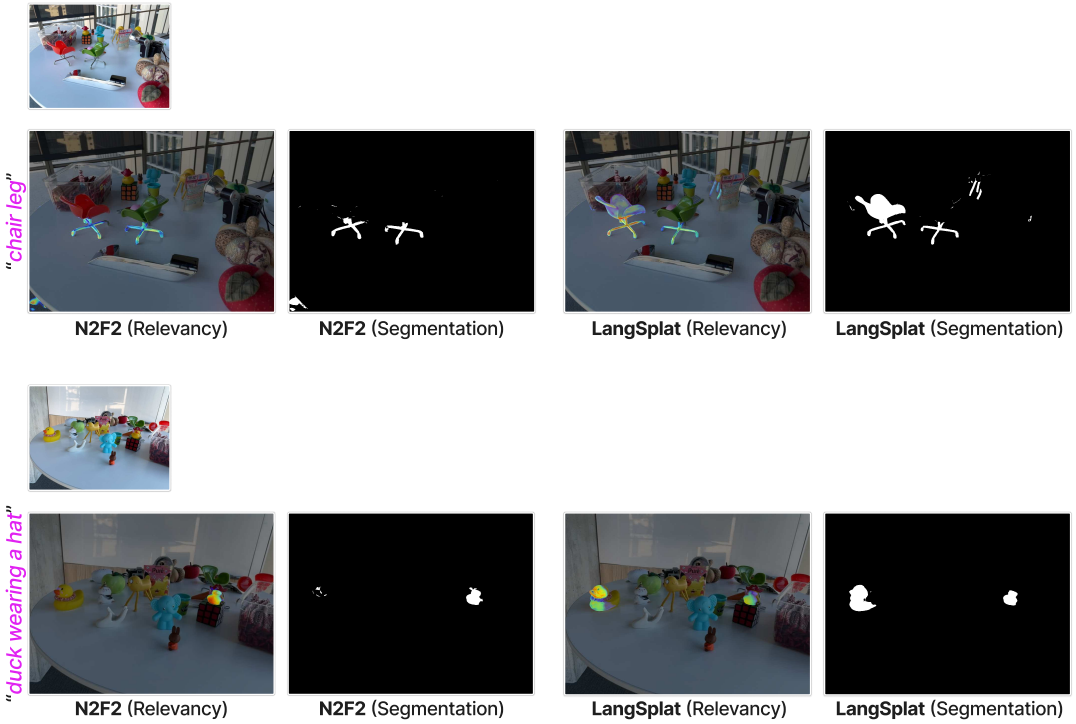


Figure 4.7: Scene: *figurines*. Each row contains results for the text query shown on the **left**. **Columns**: N2F2 (Relevancy and Segmentation maps), LangSplat [Qin et al. 2023] (Relevancy and Segmentation maps).

Chapter 5

3D-Aware Instance Segmentation and Tracking in Egocentric Videos

The paper was published at the Asian Conference on Computer Vision (ACCV), 2024.

3D-Aware Instance Segmentation and Tracking in Egocentric Videos

Yash Bhalgat[*1] Vadim Tschernezki[*1,2] Iro Laina[1]
João F. Henriques[1] Andrea Vedaldi[1] Andrew Zisserman[1]
[1]Visual Geometry Group, University of Oxford
[2]NAVER LABS Europe
{yashsb,vadim,iro,joao,vedaldi,az}@robots.ox.ac.uk

May 4, 2026

Abstract

Egocentric videos present unique challenges for 3D scene understanding due to rapid camera motion, frequent object occlusions, and limited object visibility. This paper introduces a novel approach to instance segmentation and tracking in first-person video that leverages 3D awareness to overcome these obstacles. Our method integrates scene geometry, 3D object centroid tracking, and instance segmentation to create a robust framework for analyzing dynamic egocentric scenes. By incorporating spatial and temporal cues, we achieve superior performance compared to state-of-the-art 2D approaches. Extensive evaluations on the challenging EPIC Fields dataset demonstrate significant improvements across a range of tracking and segmentation consistency metrics. Specifically, our method outperforms the next best performing approach by 7 points in Association Accuracy (AssA) and 4.5 points in IDF1 score, while reducing the number of ID switches by 73% to 80% across various object categories. Leveraging our tracked instance segmentations, we showcase downstream applications in 3D object reconstruction and amodal video object segmentation in these egocentric settings.

[*]Equal contribution.

5.1 Introduction

Egocentric videos, which capture the world from a first-person perspective, are a focus of increasing attention in computer vision due to their importance in applications such as augmented reality and robotics. Among various tools for video analysis, object tracking is of particular importance, but also faces significant challenges, in the egocentric case. Most video object segmentation (VOS) methods [H. K. Cheng et al. 2023; Siyuan Li et al. 2024; Meinhardt et al. 2022; J. Wu et al. 2022], in fact, assume that the videos contain slow, steady camera motions that keep the view centered on the object of interest [Caelles et al. 2019; Perazzi et al. 2016; Athar et al. 2023]. In comparison, egocentric videos are taken from a first-person perspective, where the camera wearer's movements introduce rapid and unpredictable changes in viewpoint. Additionally, objects frequently move in and out of the field of view, and thus are often partially or wholly occluded and/or truncated.

For example, in the EPIC KITCHENS dataset [Damen et al. 2021], the person recording the video might move a *pan* on top of a hob and leave it there for several minutes while moving around in the kitchen. During that time, they might observe more objects that look similar to the pan, which may cause an algorithm to incorrectly associate them to the pan itself. In general, video segmenters tend to lose track of the object partially or entirely due to occlusion or truncation. These issues are exacerbated when tracking multiple objects simultaneously.

Existing state-of-the-art video object segmenters try to overcome these limitations by aligning segments with dense or sparse correspondences. These are obtained from optical flow or point tracking [Rajič et al. 2023] and serve as a proxy for spatial reasoning. However, these methods can establish correspondences only in relatively short video windows due to their computational cost and poor reliability during severe viewpoint changes. The result are fragmented and incomplete object tracks, which limit their usefulness, particularly in egocentric videos.

In order to address these shortcomings, we can look at how humans locate objects. An important cue that helps correct reassociation is *object permanence*, a concept that human infants develop very early [Santrock 2002]. Permanence captures the idea that objects do not cease to exist when they are not visible. Combined with

spatial awareness, this means that the 3D location of objects at rest should not change when they are out of view or occluded. It has previously been explored for egocentric videos in 'Out of Sight, Not Out of Mind' (OSNOM) [Plizzari et al. 2024].

This brings us to the question of how to incorporate such spatial awareness in an object tracking algorithm. We achieve this by extracting scene geometry from the video stream and using it as an additional supervisory signal to refine tracks produced by a video segmentation model. More specifically, we obtain depth maps and camera parameters for the frames of the video and use this information to calculate the 3D location of the object instances. We then propose a novel approach for refining instance segmentation and tracking in egocentric videos that leverages 3D awareness to overcome the limitations of 2D trackers. By integrating a scene-level 3D reconstruction, coarse 3D point tracking, and 2D segmentation, we obtain a robust framework for analyzing dynamic egocentric videos. In particular, by incorporating both spatial *and* temporal cues from the 3D scene, our method handles occlusions and re-identifies objects that have been out of sight for some time, leading to more consistent and longer object tracks.

Our experiments on the challenging EPIC Fields dataset [Tschernezki et al. 2023] demonstrate significant improvements in tracking accuracy and segmentation consistency compared to state-of-the-art video object segmentation approaches. Furthermore, we showcase the potential of our method in downstream applications such as 3D object reconstruction and amodal video object segmentation, where the consistent and accurate object tracks produced by our method enable more accurate and complete reconstructions.

5.2 Related Work

Video object segmentation.

Video object segmentation (VOS) has seen significant advancements over the past decade [T. Zhou et al. 2022], driven by the need to accurately segment and track objects across video frames. Traditional methods often relied on frame-by-frame processing, which struggled with maintaining consistent object identities over long sequences. Early approaches such as MaskTrack R-CNN [Linjie Yang et al. 2019]

and FEELVOS [Voigtlaender et al. 2019] introduced the concept of using temporal information to improve segmentation consistency. MaskTrack R-CNN extended Mask R-CNN to video by adding a tracking head that links instances across frames, while FEELVOS utilized a pixel-wise matching mechanism to propagate segmentation masks. The introduction of memory networks and attention mechanisms marked a significant leap in performance. STM [Oh et al. 2019], AOT [Z. Yang et al. 2021] and XMem [H. K. Cheng and Schwing 2022] leveraged memory networks to store and retrieve information across frames, enabling more robust handling of occlusions and reappearances. Many recent works [Choudhuri et al. 2023; Choudhuri et al. 2021; Qiao et al. 2021; Y. Wang et al. 2021] have proposed end-to-end approaches for video object segmentation as well as panoptic segmentation. VisTR [Y. Wang et al. 2021] and SeqFormer [J. Wu et al. 2022] employed transformers to model long-range dependencies and global context. VisTR treated video segmentation as a direct set prediction problem, while SeqFormer introduced a sequential transformer architecture that processes video frames in a temporally coherent manner.

Additionally, methods like DEVA [H. K. Cheng et al. 2023] employed decoupled video segmentation approaches, combining image-level segmentation with bidirectional temporal propagation to handle diverse and data-scarce environments effectively. This also helps tackle open-vocabulary settings. MASA [Siyuan Li et al. 2024] uses the Segment Anything Model (SAM) as a robust segment proposer, and learns to match segments that correspond to the same object. An adapter can be trained to map those segments to a closed set of classes, in zero-shot settings.

Point tracking-based methods.

Point tracking-based methods have been pivotal in advancing VOS by providing a means to establish correspondences across frames. Many powerful point trackers have been recently proposed such as TAP-Vid [Doersch et al. 2022] benchmark that focused on tracking physical points in a video and works such as CoTracker [Karaev et al. 2023] and PIP [Harley et al. 2022]. CenterTrack [X. Zhou et al. 2020] combined object detection with point tracking, leveraging the strengths of both approaches. TAPIR [Doersch et al. 2023] trains an initial matching network (analogous to SeqFormer) and an iterative refinement network (which focuses

on continuous adjustments to predicted points' positions), using synthetic data, to predict accurate point tracks. SAM-PT [Rajič et al. 2023] is a point-centric interactive video segmentation method, which propagates a sparse set of points, chosen by a user, to other frames.

3D-informed instance segmentation and tracking.

A recent line of work closely related to the problem we address here involves lifting and fusing inconsistent 2D labels or segments into 3D models. In particular, Panoptic Lifting [Siddiqui et al. 2023a], ContrastiveLift [Bhalgat et al. 2023], PVLFF [H. Chen et al. 2024], and Gaussian Grouping [Ye et al. 2024] employ mechanisms for 3D instance segmentation in *static* scenes.

Operating under the assumption that objects remain stationary, they show that a 3D reconstruction of the scene enables the fusion of unassociated 2D instances (i.e. inconsistent instance identities across views) using Hungarian matching [Siddiqui et al. 2023a], contrastive learning [Bhalgat et al. 2023; H. Chen et al. 2024] or video object tracking [Ye et al. 2024; H. K. Cheng et al. 2023]. Instead of instance segmentation and tracking, GARField [C. M. Kim et al. 2024b], OmniSeg3D [Ying et al. 2024], and N2F2 [Bhalgat et al. 2024] focus on 3D hierarchical grouping, a problem which also requires resolving ambiguities that arise when fusing conflicting multi-view masks (such as those obtained by the Segment Anything Model [Kirillov et al. 2023]).

Exploiting 3D information in egocentric videos has been less explored due to the challenges of reconstructing dynamic objects. Following [Bhalgat et al. 2023], EgoLifter [Gu et al. 2024] uses contrastive learning to lift 2D segmentations to 3D, while also using a transient prediction network to handle dynamic objects. Plizzari *et al.* [Plizzari et al. 2024] focus specifically on 3D tracking of *dynamic* objects, rather than segmenting or reconstructing them. They form 3D centroid tracks by lifting 2D centroids to 3D and matching observations based on 3D distance and visual similarity. We follow [Plizzari et al. 2024], in that we lift objects to 3D using estimated depth, and initialise, match and update tracks based on 3D location and DINOv2 [Oquab et al. 2023] feature similarity. However, we also incorporate instance and category information from a base VOS model into our cost formulation, creating a more robust 3D-aware object tracking system that excels

in refining imperfect or noisy input 2D object tracks, achieving superior long-term object consistency as compared to existing 2D tracking methods.

5.3 Method

Given an egocentric video, our objective is to obtain long-term consistent object tracks by leveraging 3D information as well as an initial set of object segments and tracks obtained from a 2D-only video object segmentation (VOS) model. Our proposed method overcomes the limitations of 2D VOS models in maintaining *long-term consistent* object identities in egocentric scenarios and produces object tracks that persist despite severe occlusion and objects intermittently moving out of sight.

Fig. 5.1 provides a high-level overview of the method. We take as input an initial set of image-level segments and object tracks obtained from a pretrained VOS model. Then, we lift these 2D segments into 3D using per-frame depth from a pretrained depth estimator along with scene geometry information, and link them across time using our proposed tracking cost formulation. We first define the above problem statement more concretely in Section 5.3.1, and the 3D-aware tracking algorithm in Section 5.3.2. Then, we describe our design that includes different attributes we extract for the 2D segments in Section 5.3.3, followed by our cost formulation in Section 5.3.4.

5.3.1 Problem statement

We begin with an egocentric video sequence consisting of N frames $\mathbf{I}^t$, where $t \in \{1, \ldots, N\}$, along with the output of an off-the-shelf 2D VOS model. The objective of the method is to compute a set of tracks for the entire video $\{\mathbf{T}_i^N\}$ with associated segment IDs $\{\tilde{s}_i^N\}$ that have the desired temporal consistency. The initial output contains a set of object tracks that, while partially correct, often contain errors - particularly when objects temporarily leave the field of view or are occluded. Our goal is to *refine* and *reassemble* these tracks, leveraging 3D information to correct errors and achieve more consistent long-term tracking. Crucially, we don't discard the initial track IDs obtained from the 2D-only VOS model. Instead, we incorporate this information into our refinement process, using it as a

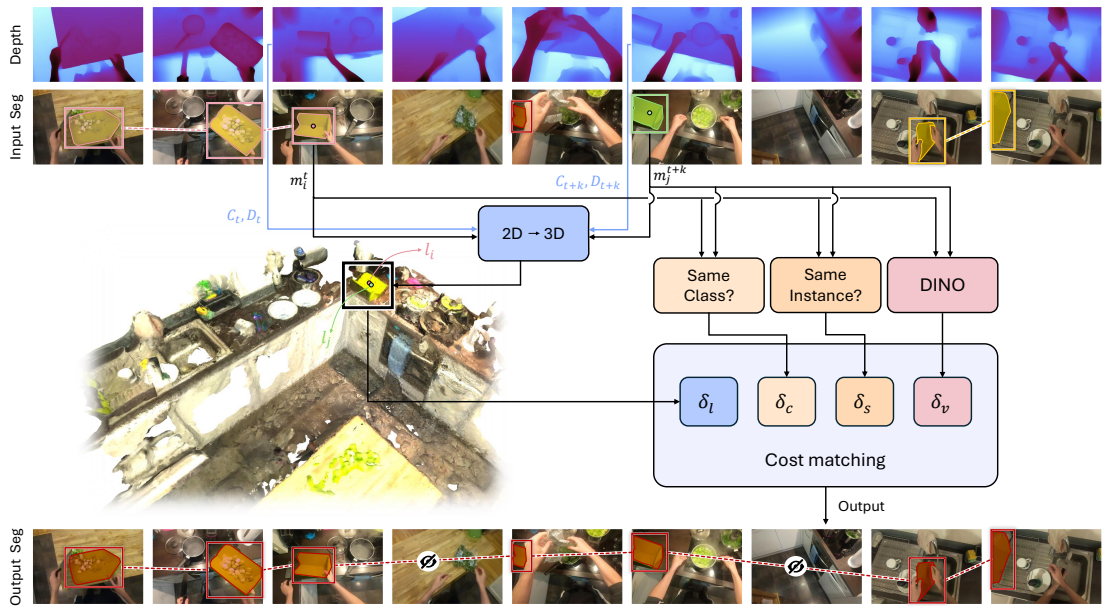


Figure 5.1: Overview of the proposed method for 3D-aware object tracking in egocentric videos. The method begins by taking image-level segments and object tracks from a pre-trained video object segmentation model, which are then lifted to 3D using per-frame depth estimates and scene geometry. These segments are fused across time with a 3D-aware tracking cost formulation to refine and maintain consistent object identities throughout the video sequence, even when the objects go out of sight (indicated by 🙈).

valuable prior for maintaining object identities. In this manner, we go beyond the previous 3D aware matching, initialisation and matching method [Plizzari et al. 2024] that we build upon.

5.3.2 3D aware tracking

First, we decompose the initial tracks into per-frame segments $\mathbf{M}^t = \{m_i^t \mid 1 \leq i \leq |\mathbf{M}^t|\}$. Specifically, each $\mathbf{M}^t$ contains a set of 2D segments m_i^t, representing the objects detected in frame t. For each segment m_i^t, we compute an attribute vector $\mathbf{b}_i^t = (b_{i,1}^t, b_{i,2}^t, \ldots, b_{i,n}^t)$ that encodes various characteristics of the segment including its initial ID s_i^t from the 2D VOS model, 3D location, visual features, and category information. These attribute vectors play a crucial role in our method, as they allow us to establish correspondences between segments across frames.

We employ a frame-by-frame track refinement approach using the Hungarian algorithm. At each frame t, we consider the existing tracks $\mathbf{T}^{t-1}$ formed in the previous $t-1$ frames and new segments $\mathbf{M}^t$ from the current frame t. The i-th track within $\mathbf{T}^{t-1}$ is associated with an attribute vector $\widetilde{\mathbf{b}}_i^{t-1}$, computed as an aggregate of the attributes of segments assigned to it (*c.f.* Section 5.3.3), and *refined* segment ID

$\tilde{s}_i^{t-1}$. We match the new segments at time t to the tracks $\mathbf{T}^{t-1}$ by solving the following optimization problem to obtain the *new refined* segment IDs $\{s_i^t\}$:

$$\arg\min_{\{s_i^t\}} \sum_{i,j} J(s_i^t, \tilde{s}_j^{t-1}, \mathbf{b}_i^t, \tilde{\mathbf{b}}_j^{t-1}) \tag{5.1}$$

subject to $s_i^t \in \{1, \ldots, S\}$ and $s_i^t \neq s_j^t$ if $i \neq j$, where S is the total number of unique object identifiers. The second condition enforces that no two segments in the same frame can have the same identifier. The cost function J is defined as:

$$J(s_i^t, \tilde{s}_j^{t-1}, \mathbf{b}_i^t, \tilde{\mathbf{b}}_j^{t-1}) = \mathbf{1}(s_i^t = \tilde{s}_j^{t-1}) \cdot \sum_{p=1}^{n} \delta_p(b_{i,p}^t, \tilde{b}_{j,p}^{t-1}) \tag{5.2}$$

Here, $\mathbf{1}(s_i^t = \tilde{s}_j^{t-1})$ is an indicator function. $\delta_p(b_{i,p}^t, \tilde{b}_{j,p}^{t-1})$ is the consistency cost for the p-th attribute between segment m_i^t in frame t and track T_j^{t-1}. Importantly, one of these δ_p functions specifically accounts for the initial track IDs (*c.f.* Eq. 5.8), encouraging our optimization to maintain these associations when appropriate.

We use the Hungarian algorithm to solve for the new segment IDs and update the initial segment IDs only if the optimisation cost from Eq. 5.2 is below a cost threshold γ. This ensures that our algorithm does not change associations when the cost is too high. Notably, for new observations that don't match any existing track (i.e., their matching cost exceeds γ), we initialize new tracks. Importantly, we do not terminate tracks that fail to match with a new observation in the current frame. Instead, we maintain these tracks in our database, propagating their attributes from time $t-1$ to time t. This approach allows our method to handle temporary occlusions or brief disappearances of objects, maintaining object identity over longer periods.

By iteratively applying this process across the entire video sequence, we refine the initial tracks, correcting errors while still leveraging the valuable information provided by the 2D VOS model. Our method's ability to incorporate both the initial 2D tracking information and additional 3D cues, combined with its frame-by-frame processing and track maintenance strategy, enables it to effectively handle the challenges of egocentric videos, including frequent occlusions, objects moving in and out of view, and rapid camera motion. Next, we describe how we define and compute the segment attributes b_i^t as well as the associated cost functions δ_p.

5.3.3 Attributes for 3D-aware cost formulation

Our method leverages 3D information to improve the *initial* object tracks obtained from an off-the-shelf 2D-only VOS model. In addition to 3D location information, we leverage appearance information (visual features), as well as categorical information (i.e. the initial category and instance labels from the 2D model) to refine the segment associations. We denote the attributes for each segment as $\mathbf{b}_i^t = (l_i^t, v_i^t, c_i^t, s_i^t)$, where l_i^t is the 3D location of the segment, v_i^t is the visual feature, c_i^t is the category label and s_i^t is the instance label.

3D locations as segment attributes. We are given for each image $\mathbf{I}^t, t \in \{1, \dots, N\}$, a camera pose $\mathbf{C}^t$, camera intrinsics K and a depth map $\mathbf{D}^t$. In order to optimise the associations with 3D information, we lift the 2D centroid of each segment into 3D. We define the 3D centroid of segment m_i^t in frame t as l_i^t, representing one out of several attributes of $\mathbf{b}_i^t$. We calculate the location of this segment by projecting its 2D centroid into 3D with

$$l_i^t = \mathbf{C}^t \begin{bmatrix} d_i^t K^{-1} \begin{bmatrix} x_i^t, & y_i^t, & 1 \end{bmatrix}^T \\ 1 \end{bmatrix}, \tag{5.3}$$

where d_i^t is the depth value obtained from $\mathbf{D}^t$ that corresponds to the centroid of segment m_i^t of frame t, and x_i^t, y_i^t are the 2D coordinates of the centroid.

Visual features as segment attributes. While the 3D location of a segment plays a crucial role in overcoming the mentioned problems of associating segments throughout occlusions, viewpoint changes and similar issues, we also make use of 2D-level visual features v_i^t as one of the attributes $\mathbf{b}_i^t$ that correspond to each segment. Specifically, for an image $\mathbf{I}^t$ and each segment m_i^t of the image, we use a pretrained vision encoder, e.g. DINOv2 [Oquab et al. 2023], to obtain the visual feature v_i^t as:

$$v_i^t = V(\text{crop}(\mathbf{I}^t \odot m_i^t)), \tag{5.4}$$

where V is the vision encoder and $\odot$ denotes Hadamard product. The 'crop' operation extracts the smallest patch with a 1:1 aspect ratio enclosing mask m_i^t.

Initial instance and category labels as segment attributes. Our proposed method refines the *initial* tracks obtained from a purely 2D video object segmentation model. Let $\bar{c}_i^t$ and $\bar{s}_i^t$ denote the initial category and instance labels for segment m_i^t obtained from the 2D model. We use $\bar{c}_i^t$ as an attribute to discourage the optimisation from matching instances which did not initially belong to the same category. And similarly, we use $\bar{s}_i^t$ to encourage the optimization to preserve the initial tracks of instances across frames obtained from the 2D model. We mathematically define the associated costs below.

Attributes for a track. A track $\mathbf{T}^{t-1}$ that exists at time $t-1$ is a sequence of segments assigned to it so far. We associate each track with an attribute vector $\widetilde{\mathbf{b}}_i^{t-1} = (\tilde{l}_i^t, \tilde{v}_i^t, \tilde{c}_i^t, \tilde{s}_i^t)$, where $\tilde{l}_i^t$, $\tilde{c}_i^t$ and $\tilde{s}_i^t$ are defined to be the corresponding attributes of the most recent segment assigned to this track. The visual feature attribute $\tilde{v}_i^t$ is defined to be the mean visual feature of the 100 most recent segments assigned to the track.

5.3.4 Cost functions

The attributes used for refining the tracks are thus $\mathbf{b}_i^t = (l_i^t, v_i^t, \bar{c}_i^t, \bar{s}_i^t)$, consisting of the 3D location, the visual features, *initial* category label and *initial* instance label for the segment m_i^t of frame t. Now, we define the cost functions δ_p used in Eq. 5.2 for these individual attributes. We follow [Rajasegaran et al. 2022; Plizzari et al. 2024] for the first two:

1. We model the 3D location cost δ_l with the exponential distribution as follows:

$$\delta_l(l_i^t, l_j^{t'}) = -\log\left(\frac{1}{\alpha_l}\exp\left(-\|l_i^t - l_j^{t'}\|_2\right)\right) \tag{5.5}$$

2. We model the cost for the visual features, δ_v, using a Cauchy distribution:

$$\delta_v(v_i^t, v_j^{t'}) = -\log\left(\frac{1}{1+\alpha_v\|v_i^t - v_j^{t'}\|_2^2}\right) \tag{5.6}$$

3. For the category and instance label, we use a $0-1$ cost function and refer to it with δ_c and δ_s:

$$\delta_c(\bar{c}_i^t, \bar{c}_j^{t'}) = \begin{cases} 0 & \text{if } \bar{c}_i^t = \bar{c}_j^{t'} \\ \alpha_c & \text{if } \bar{c}_i^t \neq \bar{c}_j^{t'} \end{cases}, \quad (5.7) \qquad \delta_s(\bar{s}_i^t, \bar{s}_j^{t'}) = \begin{cases} 0 & \text{if } \bar{s}_i^t = \bar{s}_j^{t'} \\ \alpha_s & \text{if } \bar{s}_i^t \neq \bar{s}_j^{t'} \end{cases} \quad (5.8)$$

Here, α_l, α_v, α_c and α_s are used to modulate the importance of each cost function. The cost parameters for the category and instance labels discourage the matching of segments that are inconsistent with the category and instance labels from the input segments. As described in Section 5.3.2, we consider the tracks formed in previous $t-1$ frames and match them to the new observations from the current frame t using the Hungarian algorithm.

We refer the reader to Section 5.6.1 for the implementation details and to Section 5.6.1 for hyperparameter settings.

5.4 Experiments

5.4.1 Benchmark and baselines

We evaluate our proposed method on 20 scenes from the EPIC Fields [Tschernezki et al. 2023] dataset. EPIC Fields comprises of complex real-world videos with a high diversity of activities and object interactions, making it an ideal testbed for our evaluation. The selected videos include varied lighting conditions, occlusions, objects that disappear from sight, and have an average length of 10 minutes. To further demonstrate our method's capability, we also evaluate it on the Ego4D [Grauman et al. 2022] dataset and report the results in Table 5.5.

We compare against the following baselines:

1. **DEVA** [H. K. Cheng et al. 2023] employs a decoupled video segmentation approach that combines task-specific image-level segmentation with a class-agnostic bi-directional temporal propagation model. This method is particularly effective in diverse and data-scarce environments, as it separates image and video segmentation tasks to improve overall tracking accuracy by reducing the impact of image segmentation errors.

2. **MASA** [Siyuan Li et al. 2024] is a more recent state-of-the-art method that focuses on robust instance association learning. MASA includes a universal

adapter that allows it to integrate with various foundational segmentation or detection models, enhancing its ability to track any detected objects robustly. By utilizing features from these underlying 2D models, MASA can improve the instance and category assignments, providing robust zero-shot tracking capabilities in complex domains.

Note that, both DEVA and MASA can be used with various 2D object detection models. We tested both methods with three 2D models: OWLv2 [Matthias Minderer 2023], Detic [X. Zhou et al. 2022] and GroundingDINO [S. Liu et al. 2023], and found that DEVA works best with OWLv2 while MASA works best with Detic on the EPIC Fields dataset. Hence, we incorporate **DEVA + OWLv2** and **MASA + Detic** as baselines in our experiments.

Since both baselines use an open-vocabulary 2D detection model, we use text prompts corresponding to the object categories from EPIC Fields [Tschernezki et al. 2023] to obtain image-level object bounding boxes (with associated class labels).

In addition to these, we also simulate the OSNOM [Plizzari et al. 2024] approach within our framework and provide a detailed comparison in Section 5.6.5.

5.4.2 Metrics

We evaluate our method using the HOTA (Higher Order Tracking Accuracy) metric [Luiten et al. 2021]. HOTA assesses multi-object tracking (MOT) performance by combining detection accuracy (DetA), association accuracy (AssA), and localization IoU (Loc-IoU). It is calculated as the geometric mean of DetA and AssA over various Loc-IoU thresholds α:

$$\text{HOTA} = \frac{1}{|S|} \sum_{\alpha \in S} \text{HOTA}(\alpha) = \frac{1}{|S|} \sum_{\alpha \in S} \sqrt{\text{DetA}(\alpha) \times \text{AssA}(\alpha)}$$

where S is the set of IoU thresholds. We use $S = \{0.05, 0.1, \ldots, 0.9, 0.95\}$ following standard protocol [Luiten et al. 2021]. DetA measures the overlap between the set of *all* predicted segments and *all* ground-truth (GT) segments. It is defined as:

$$\text{DetA}(\alpha) = \frac{|\text{TP}_\alpha|}{|\text{TP}_\alpha| + |\text{FP}_\alpha| + |\text{FN}_\alpha|}$$

True Positives (TP_α) are identified by matching predicted segments to GT segments with an IoU $\geq \alpha$ using Hungarian matching. Unmatched predictions are False Positives (FP_α), and unmatched GT segments are False Negatives (FN_α).

AssA measures the tracker's ability to maintain consistent object identities over time:

$$\mathrm{AssA}(\alpha) = \frac{1}{|\mathrm{TP}_\alpha|} \sum_{c \in \mathrm{TP}_\alpha} \frac{\mathrm{TPA}(c)|}{|\,\mathrm{TPA}(c)| + |\,\mathrm{FPA}(c)| + |\,\mathrm{FNA}(c)|}$$

where we iterate over all TP pairs, measuring the alignment between the predicted and ground-truth segment's *whole* track. True Positive Associations (TPA) represents the number of TP matches between the two chosen tracks for a pair.

Additionally, we use the IDF1 (Identity F1) score to measure how well the tracker maintains consistent object identities throughout the sequence:

$$\mathrm{IDF1} = \frac{2\,|\mathrm{IDTP}|}{2\,|\mathrm{IDTP}| + |\mathrm{IDFP}| + |\mathrm{IDFN}|}$$

where IDTP (Identity True Positives) represents matches on overlapping parts of tracks that are matched, while IDFP (Identity False Positives) and IDFN (Identity False Negatives) represent the remaining GT and predicted segments.

5.4.3 Results

We evaluate our method against DEVA [H. K. Cheng et al. 2023] and MASA [Siyuan Li et al. 2024] using the HOTA, DetA, AssA, and IDF1 metrics. Table 5.1 presents the overall results as well as scene-specific performance. Fig. 5.2 provides a qualitative comparison of results.

Our approach consistently outperforms both baselines across various metrics. Compared to DEVA, our method achieves an overall HOTA score of 27.72, a notable improvement over DEVA's 25.14. This enhancement is even more pronounced in the AssA metric, which measures the tracker's ability to maintain consistent object identities over time. Our method attains an AssA score of 43.90, substantially higher than DEVA's 36.72.

This further underscores our method's superior performance in maintaining consistent object identities throughout the video sequences. Our method also shows significant improvements in IDF1 scores, achieving 26.63 compared to DEVA's 22.17.

Table 5.1: Results on the EPIC Fields [Tschernezki et al. 2023] dataset.

Video	DEVA [H. K. Cheng et al. 2023]				Ours (w/ DEVA)				MASA [Siyuan Li et al. 2024]				Ours (w/ MASA)			
	HOTA	DetA	AssA	IDF1	HOTA	DetA	AssA	IDF1	HOTA	DetA	AssA	IDF1	HOTA	DetA	AssA	IDF1
P01_01	33.60	25.25	45.68	28.61	41.91	24.94	71.85	38.76	9.11	4.64	17.99	8.15	8.36	4.64	15.12	7.50
P01_104	25.79	22.98	29.09	21.93	30.92	22.95	41.88	31.40	11.66	8.81	15.59	9.61	12.77	8.81	18.64	10.54
P02_09	30.07	21.85	42.29	23.46	33.76	21.77	53.11	27.73	20.51	15.46	27.46	17.67	19.04	15.47	23.68	16.45
P02_121	8.75	7.47	10.32	6.07	11.79	6.64	20.96	12.09	6.71	5.68	8.03	4.06	9.29	5.69	15.34	6.90
P02_132	26.71	25.05	28.80	29.04	29.96	24.74	36.56	35.18	15.44	11.40	21.28	13.31	15.39	11.35	20.98	13.65
P03_101	27.56	21.07	36.13	24.17	29.63	19.72	44.61	26.67	7.71	6.22	9.65	4.55	9.53	6.22	14.76	6.97
P04_03	15.60	11.72	23.41	11.24	16.85	11.64	26.63	12.23	10.21	5.12	22.80	6.22	10.17	5.12	21.92	6.67
P04_11	43.03	35.83	52.05	48.88	43.13	35.87	52.21	49.74	10.82	7.26	16.30	11.26	10.54	7.27	15.37	9.76
P04_25	18.71	6.02	58.25	10.39	18.71	6.18	56.79	10.64	12.64	5.96	27.30	6.69	13.46	5.96	30.45	8.54
P06_01	26.22	23.80	29.60	28.12	29.73	25.65	34.95	35.05	18.95	21.33	19.02	17.96	26.01	21.33	32.87	30.84
P06_102	27.71	17.37	44.81	23.75	30.42	18.09	51.88	28.71	10.42	6.17	18.87	4.87	8.71	6.18	13.91	4.17
P06_12	42.47	27.00	68.89	41.40	44.13	26.95	73.86	48.41	41.94	28.57	62.01	47.70	44.35	28.54	69.14	52.38
P07_101	18.45	15.66	23.28	14.28	23.12	15.95	34.81	21.44	12.25	7.83	19.98	8.58	12.98	7.82	22.38	9.44
P11_103	27.73	15.55	49.78	24.77	24.68	15.16	40.42	21.98	11.69	8.11	17.37	7.75	13.25	8.02	21.98	9.57
P12_02	23.51	15.26	37.40	16.16	26.21	15.45	45.35	20.63	11.46	7.33	17.96	7.46	12.77	7.34	22.34	9.59
P22_117	18.15	12.62	27.06	13.50	22.06	12.34	39.90	18.92	7.46	3.32	17.88	4.63	6.29	3.33	12.37	3.97
P24_05	19.07	12.27	30.48	16.10	21.02	12.27	36.50	19.85	11.39	9.12	14.26	7.27	13.15	9.09	19.06	10.00
P28_109	24.77	17.39	35.36	21.68	25.99	18.08	37.37	26.32	12.82	11.49	14.38	9.97	13.29	11.49	15.41	10.97
P28_14	27.11	18.85	39.90	25.30	28.04	18.18	44.28	27.54	13.22	9.21	20.28	10.17	13.35	9.17	20.16	10.61
P37_101	17.85	14.97	21.79	14.56	22.23	14.93	33.96	19.24	11.09	9.26	13.86	8.54	11.20	9.14	14.07	8.76
Overall	25.14	18.40	36.72	22.17	**27.72**	18.38	**43.90**	**26.63**	13.73	10.06	20.32	10.82	**14.67**	10.04	**22.43**	**12.36**

Similar improvements are observed when comparing to MASA, which demonstrates our approach's adaptability to different base models.

Notably, DetA scores remain relatively consistent across all methods (e.g. 18.40 for MASA vs. 18.38 for our method when using MASA as the base model). This is because our method improves the instance and category assignments for the segments using 3D information but does not alter the segments themselves. Since the DetA metric only evaluates the segments regardless of IDs, it results in similar scores for both the base 2D method and our method.

Scene-specific analysis. Our method shows remarkable improvements in complex scenes, such as P01_01, where we achieve a HOTA score of 41.91 compared to DEVA's 33.60, a 24% improvement. This scene likely contains frequent object occlusions or out-of-view instances where our 3D-aware approach excels. Significant improvements are also observed in scenes like P07_101 and P22_117, with improvements of 25% and 22% respectively in HOTA scores.

The AssA metric shows the most significant improvements. For example, in P02_121, our method achieves an AssA of 20.96 compared to DEVA's 10.32, a 103% improvement. However, the degree of improvement varies across scenes. In some, like P04_11, the improvement is marginal, suggesting that not all scenes benefit equally from 3D awareness.

Table 5.2: Number of ID switches averaged over all videos, shown for challenging and frequently appearing objects. Last column: number of videos featuring each object.

Object Class	DEVA [H. K. Cheng et al. 2023]	Ours (w/ DEVA)	# videos
tap	14.53	**2.88**	17
knife	27.21	**5.29**	14
chopping board	20.25	**5.42**	12
spoon	21.00	**5.00**	10
bowl	23.11	**5.67**	9
pan	19.44	**4.11**	9
sponge	22.38	**5.38**	8

Analysis of ID switches by object class. To further understand our method's performance in maintaining consistent object identities, we analyze the number of ID switches occurring throughout the videos for different object categories. Table 5.2 shows the average number of ID switches over all videos for a subset of challenging and commonly occurring object classes in the EPIC Fields dataset, comparing our method to the DEVA baseline. Our approach consistently and significantly reduces ID switches across all shown object classes, with improvements ranging from 73% to 80% reduction. For instance, small objects, prone to occlusions, such as knives, see a reduction from 27.21 to 5.29 switches, taps from 14.53 to 2.88, and pans from 19.44 to 4.11. This substantial improvement across various object types, regardless of their size or frequency of appearance, demonstrates the robustness of our 3D-aware approach. It highlights our method's effectiveness in maintaining consistent object identities through complex interactions and occlusions typical in egocentric videos, particularly for frequently manipulated kitchen objects and objects that may remain stationary across time, while not necessarily staying in view.

5.4.4 Ablations

Comparison with other *plug-and-play* tracking methods The above results demonstrated our method's generalization capability by combining with with two state-of-the-art methods, DEVA [H. K. Cheng et al. 2023] and MASA [Siyuan Li et al. 2024]. To further highlight our method's versatility, we compare it with other existing *plug-and-play* tracking algorithms, namely BoTSORT [Aharon et al. 2022], ByteTrack [Y. Zhang et al. 2022], OCSORT [J. Cao et al. 2023] and

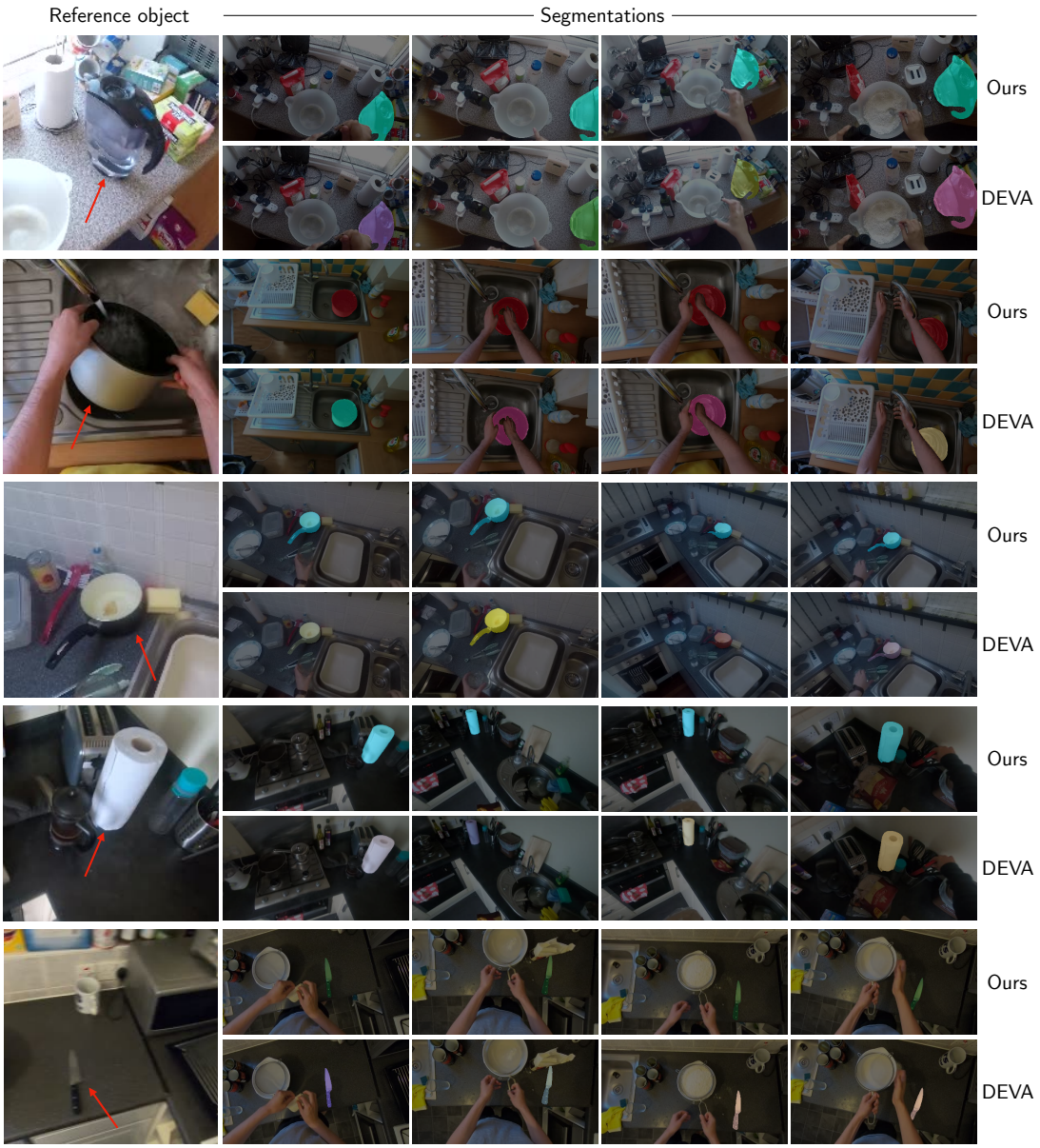


Figure 5.2: Qualitative comparison between our method and DEVA [H. K. Cheng et al. 2023]. We show instance segmentations for selected reference objects. Our method maintains consistent tracks despite viewpoint changes and objects going out of view, while DEVA's tracks break. Our approach successfully segments the pot even when in motion.

DeepOCSORT [Maggiolino et al. 2023]. We use the same ReID model with all 4 tracking methods and use OWLv2 [Matthias Minderer 2023] as the 2D segmentation model for fair comparisons. Table 5.3 shows that all four methods perform less favourably than DEVA [H. K. Cheng et al. 2023] even while using the same base 2D model, and thus are outperformed by our method which further refines the tracks from DEVA.

Table 5.3: Comparison with other plug-and-play tracking methods.

Method	HOTA	DetA	AssA	IDF1
BoTSORT [Aharon et al. 2022]	12.83	7.81	24.23	10.30
ByteTrack [Y. Zhang et al. 2022]	20.08	16.31	27.56	16.94
OCSORT [J. Cao et al. 2023]	21.90	17.94	29.28	18.95
DeepOCSORT [Maggiolino et al. 2023]	22.63	17.98	31.31	19.88
DEVA [H. K. Cheng et al. 2023]	25.14	**18.40**	36.72	22.17
Ours (w/ DEVA)	**27.72**	18.38	**43.90**	**26.63**

Table 5.4: Influence of different components in the tracking formulation.

Instance	Category	3D Location	Visual	HOTA	DetA	AssA	IDF1
✓	✓	✓	✓	27.72	18.38	43.90	26.63
✓	✓	✓	✗	27.17	18.38	42.45	26.12
✓	✓	✗	✓	26.32	18.37	41.23	26.04
✓	✗	✓	✓	25.49	18.11	38.74	24.18
✗	✓	✓	✓	25.96	18.38	39.51	24.50
✗	✓	✗	✗	21.11	18.41	26.42	16.80
✓	✓	✗	✗	25.14	18.40	36.72	22.19
DEVA [H. K. Cheng et al. 2023]				25.14	18.40	36.72	22.17

Influence of different components on tracking. Our tracking formulation consists of four components (Eq. 5.8, 5.7, 5.5 and 5.6): instance cost, category cost, 3D location cost, and visual feature cost. We evaluate the influence of each component by turning off the corresponding cost one at a time in the cost-matching formulation. Table 5.4 shows that all components contribute positively to the tracking performance, but to varying degrees. Removing the visual features has the least impact, reducing the HOTA score from 27.72 to 27.17. The 3D location information proves more important, with its removal causing the HOTA score to drop to 26.32. Removing the category term has the most significant impact on the tracking performance, followed by the instance cost. Note that, if the instance cost is removed, the cost optimization completely ignores the *initial* tracks provided by the 2D base tracker (e.g. DEVA or MASA), effectively finding instance tracks from scratch. Notably, even without this initial guidance, our method outperforms the 2D tracking method (DEVA [H. K. Cheng et al. 2023]) in terms of HOTA (+0.82), AssA (+2.79) and IDF1 (+2.33).

Metrics across IoU thresholds. As described in Section 5.4.2, HOTA, DetA, and AssA can be calculated at different IoU thresholds. Fig. 5.3 illustrates how

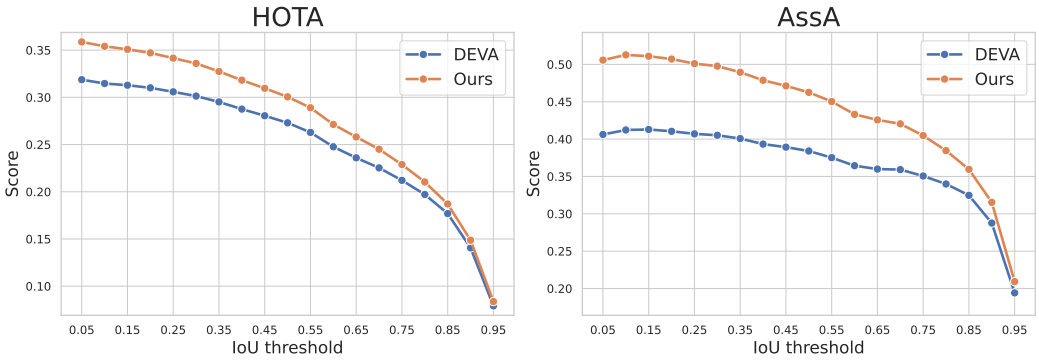


Figure 5.3: HOTA and Association accuracy (AssA) metrics across different IoU thresholds.

these metrics change as the IoU threshold increases. As expected, all metrics decrease with higher thresholds, as stricter overlap requirements lead to fewer True Positive matches between predicted and ground-truth segments. Notably, our method consistently outperforms DEVA across all thresholds for both HOTA and AssA metrics, while he AssA curve shows a more pronounced improvement. This suggests that our 3D-aware approach is particularly effective at maintaining consistent object identities throughout the video sequence, even under strict evaluation criteria.

5.5 Conclusion

In this paper, we presented a novel 3D-aware approach to instance segmentation and tracking in egocentric videos, addressing the unique challenges of first-person perspectives. By integrating 3D information, our method significantly improves tracking accuracy and segmentation consistency compared to state-of-the-art 2D approaches, especially over long periods. Our ablation studies highlight the importance of 3D information and the category as well as instance cost terms in matching, while also showing robustness to hyperparameter changes. Beyond improved tracking, our approach enables valuable downstream applications such as high-quality 3D object reconstructions and amodal segmentation. This work demonstrates the power of incorporating 3D awareness into egocentric video analysis, opening up new possibilities for robust object tracking in challenging first-person scenarios.

Acknowledgements. We're funded by EPSRC AIMS CDT EP/S024050/1 and AWS (Y. Bhalgat), NAVER LABS Europe (V. Tschernezki), Royal Academy of Engineering RF\201819\18\163 (J. Henriques), ERC-CoG UNION 101001212 (A. Vedaldi and I. Laina), and EPSRC VisualAI EP/T028572/1 (I. Laina, A. Vedaldi and A. Zisserman). We thank Chiara Plizzari for sharing evaluation details of the OSNOM baseline, and Ahmad Dar Khalil for helping with the annotations for our evaluation.

5.6 Appendix

5.6.1 Implementation details

In the EPIC Fields [Tschernezki et al. 2023] dataset, the per-frame camera pose and intrinsics are obtained using COLMAP [Schonberger and Frahm 2016], which also provides a sparse point cloud representing the *static* parts of the scene. We follow [Plizzari et al. 2024], and obtain the depth maps, $\mathbf{D}_t$, by first calculating a mesh from the sparse point clouds, and then aligning the predictions of a pre-trained depth model to the mesh through shift and scale transformations. We use Depth Anything [Lihe Yang et al. 2024], a state-of-the-art monocular depth estimation model. We obtain the scale-shift parameters for each depth map by optimizing the L1 distance between the transformed depth map and the rasterized mesh depth. We use the DINOv2 [Oquab et al. 2023] encoder to compute the visual features for segments. When optimizing the tracking cost, the visual feature for a "track" is computed as the average of the visual features of the most recent 100 observations assigned to the track.

Details on frame-by-frame Tracking Cost optimization

Our tracking algorithm processes the video sequentially, applying the cost optimization frame-by-frame. At each frame t, we consider:

1. M existing tracks from the previous $t-1$ frames
2. N new observations from the current frame t

Here, an observation refers to the set of attributes for a segment (Section 3.2 of main paper), while a track is a sequence of observations that have been assigned to

the same instance across frames. We employ the Hungarian algorithm to perform matching between the M existing tracks and N new observations. This matching process is guided by our cost formulation (Section 3.3).

For new observations that do not match any existing track (i.e. heir matching cost exceeds the threshold γ), we initialize new tracks. This allows our method to accommodate the introduction of new objects into the scene.

Importantly, we do not terminate tracks that fail to match with a new observation in the current frame. Instead, we maintain these tracks in our database, propagating their attributes from time $t-1$ to time t. This approach allows our method to handle temporary occlusions or brief disappearances of objects, maintaining object identity over longer periods.

This process enables effective tracking of multiple objects across extended video sequences, addressing challenges like object entries, exits, and occlusions.

Hyperparameters

Our model has five hyperparameters: γ, α_s, α_v, α_l, α_c. We set $\alpha_c = 10^4$ and $\gamma = 30$ based on observed cost values. The remaining parameters were tuned on a held-out set of 4 videos, yielding optimal values of $\alpha_s = 10, \alpha_v = 2, \alpha_l = 10$. These settings were used across all experiments.

Evaluation Data

We evaluate our method and baselines using the VISOR dataset, which provides pixel-level annotations for active objects in kitchen environments. These annotations include any objects used for cooking or cleaning. From these annotations, we derive ground truth tracks and segmentations. The dataset's annotation structure supports instance-level tracking, as segments of a particular object category often correspond to the same instance throughout a video. As mentioned in the main paper, we evaluate our approach using the VISOR annotations for 20 videos from EPIC Fields [Tschernezki et al. 2023] dataset.

5.6.2 Additional results on the Ego4D dataset

To further demonstrate our method's applicability, we also include results on a few select scenes from the Ego4D [Grauman et al. 2022] dataset. We follow the same evaluation protocol used with the EPIC Fields dataset and utilize Egotracks [H. Tang et al. 2024] for the ground-truth track annotations.

Table 5.5 shows that our method can consistently refine the tracks obtained by DEVA [H. K. Cheng et al. 2023] on 3 videos. We use the same hyperparameters for these experiments as the ones used with the EPIC Fields datasets described in Section 5.6.1. The video lengths are 2-3× shorter compared to EPIC Fields, which we believe results in smaller margins of improvement as compared to the ones showed in the main paper.

Table 5.5: Results on the Ego4D [Grauman et al. 2022] dataset.

Video ID	DEVA [H. K. Cheng et al. 2023]				Ours (w/ DEVA)			
	HOTA	DetA	AssA	IDF1	HOTA	DetA	AssA	IDF1
8b47ac19-7c4f-47d2-b5d0-755b524b66b2	15.29	12.26	22.74	13.56	**17.40**	12.22	**28.15**	**14.98**
9f5253af-acc3-40ca-b8bf-7b931f875bd7	12.37	9.25	18.61	10.43	**14.38**	9.25	**23.04**	**12.33**
bff3d583-ca3b-44b8-9740-3b34c5a8d7a9	21.58	13.60	36.21	18.27	**23.73**	13.58	**43.96**	**20.16**

5.6.3 On-device Inference Runtime Analysis

Our method processes egocentric videos in an online manner. While we compute meshes in advance, we track the objects online with the entire pipeline running at 20FPS on a A6000 GPU. Each of our method's components runs as follows: DINOv2 at 43FPS, the lifting to 3D with DepthAnythingV2 at 23FPS, the prediction of segmentation masks with OWLv2 (run every 5 frames) at 31FPS, the temporal propagation with DEVA [H. K. Cheng et al. 2023] at 25 FPS. We use `torch.cuda.stream` for asynchronous execution of all models on the same GPU.

5.6.4 Sensitivity to Hyperparameters

We evaluate the sensitivity of our method by varying the values of the 4 hyperparameters: α_s, α_c, α_l, α_v in the cost-matching formulation. We perform this analysis on a subset of 5 videos, using 3 representative values for each hyperpa-

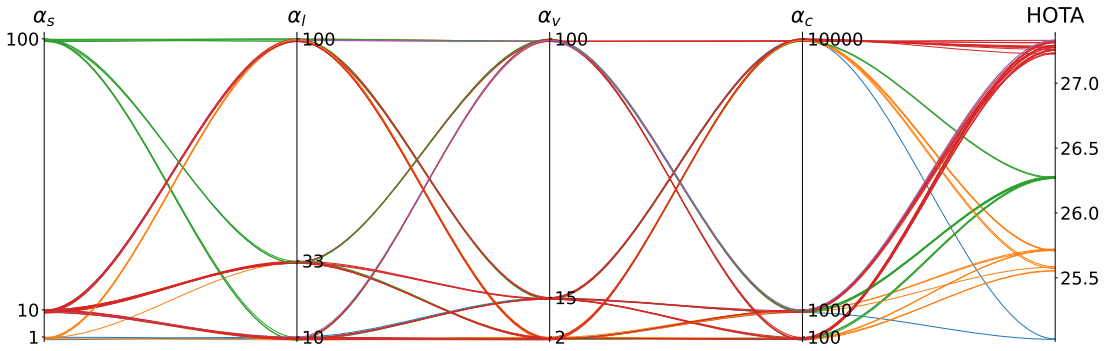


Figure 5.4: Sensitivity analysis of HOTA performance to hyperparameters. Each vertical axis represents a hyperparameter ($\alpha_s, \alpha_l, \alpha_v, \alpha_c$) or the HOTA metric (*rightmost* axis). Colored lines show individual configurations, where intersections with the vertical axes indicating parameter values and resulting HOTA scores.

rameter, resulting in $3^4 = 81$ configurations. Fig. 5.4 shows that 57 out of these 81 hyperparameter configurations lead to a HOTA score in the range 27.2 ± 0.2, which shows the robustness of our method to these parameters. There are some configurations, e.g. when $\alpha_s = 100$ or $\alpha_c = 100$, that lead to a degradation in performance.

5.6.5 Ablation without *category* and *instance* terms

Our method, although inspired by OSNOM [Plizzari et al. 2024], offers more than coarse object centroid tracking. Our method is tailored to be more suitable for long-term video *instance segmentation* by taking in account the initial categorical information provided by the base 2D model. By focusing on fine-grained instance segmentations, our approach significantly improves 2D-level tracking performance in egocentric videos. Importantly, its simplicity ensures easy *plug-and-play* functionality across different trackers, as shown in our experiments.

Here, we ablate our method by disabling the instance and category terms in our cost formulation, relying solely on 3D location and visual feature costs, which brings our method closer to OSNOM. Note that, this ablation is compared with the "full" version of our method on the object tracking task.

As shown in Table 5.6, this OSNOM-like configuration results in a significant performance drop, with metrics falling below even the initial performance obtained using the base 2D model, DEVA [H. K. Cheng et al. 2023]. This is due to two reasons. First, without the "instance" cost, the model completely ignores the

Table 5.6: Ablation of our method without *category* and *instance* terms in the cost formulation, evaluated on the egocentric object tracking task.

Method	HOTA	DetA	AssA	IDF1
DEVA [H. K. Cheng et al. 2023]	25.14	18.40	36.72	22.17
Ours (without *cat* and *ins* terms)*	11.31	6.49	24.16	12.98
Ours (full version)	27.72	18.38	43.90	26.63

initial tracks provided by DEVA. Second, without the "category" cost, the model often confuses objects across categories (e.g. pot vs sink, knife vs spoon). Since the Detection Accuracy (DetA) for a video/scene is computed on predicted instances per class, this leads to a severely low DetA on account of various misclassified instances. This comparison underscores the importance of our additional cost terms in maintaining robust long-term tracking performance in egocentric settings.

5.6.6 Downstream applications

Our 3D-aware instance segmentation and tracking method yields longer and more consistent tracks than 2D methods. This improvement enables two key downstream applications: 3D object reconstruction and amodal segmentation.

Reconstruction of objects. The longer, more consistent tracks produced by our method allow us to extract the same object from many frames using the output instance ID. This multi-view information is crucial for achieving high-quality 3D reconstructions. This is something that fragmented or inconsistent tracks from 2D methods often fail to achieve. Additionally, our 3D tracking approach, which uses lifted centroids, allows us to determine the time ranges when an object remains static. We leverage these static periods for reconstruction, as they provide the most reliable geometric information. This selective use of frames is only possible due to maintaining long-term tracks of objects.

Amodal segmentation. Amodal segmentation aims to estimate the full extent of objects, including parts that are occluded. Building upon our 3D object reconstructions, we render the reconstructed 3D object from multiple viewpoints corresponding to different frames in the video. This process allows us to generate occlusion-free, amodal segmentations of the object.

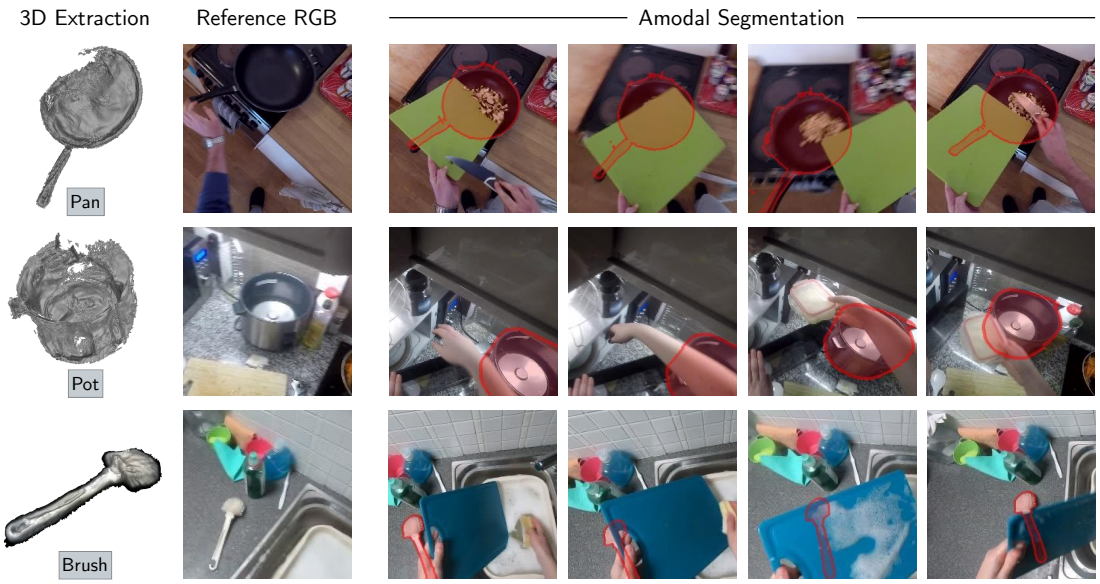


Figure 5.5: Qualitative results demonstrating the quality of object reconstructions and amodal segmentations obtained using our 3D-aware tracking method. The "Reference RGB" column show an image containing the referred object *unoccluded*. Last 4 columns show the resulting amodal segmentations of the object in red masks with a **red** border.

These applications demonstrate the cascading benefits of our improved 3D-aware tracking method. We show qualitative results in Fig. 5.5 that demonstrate the quality of object reconstructions and amodal segmentations obtained using our method. In practice, we use the 2D Gaussian Splatting [Huang et al. 2024] approach to obtain precise mesh reconstructions for these objects.

5.6.7 Details on Obtaining Amodal Segmentations

This section elaborates on the process of obtaining amodal segmentations, which involves three main steps: identifying static object frames, 3D object reconstruction, and amodal segmentation projection.

Identifying Static Object Frames: We begin by analyzing the tracked 3D centroid of the object of interest across the video sequence. By identifying periods where the centroid remains relatively stationary (using a threshold on 3D location differences between frames), we can isolate a range of frames where the object is static. This step is crucial as it allows us to gather multiple views of the object from different camera angles while minimizing the complexity introduced by object motion.

3D Object Reconstruction: Once we have identified the static frames, we utilize the corresponding 2D instance segmentations and associated camera parameters to reconstruct the 3D shape of the object. This reconstruction is achieved through a technique known as Gaussian Splatting[1]. In this approach, we represent the 3D object as a collection of Gaussian functions in 3D space. Each Gaussian is characterized by its mean position and covariance matrix, which define its location and shape respectively. Given G as the set of 3D Gaussians and a camera viewpoint C_i, the differentiable Gaussian Splatting renderer [Kerbl et al. 2023] produces an image

$$\hat{I} = \Pi(G, C_i) \in \mathbb{R}^{H \times W \times 3}$$

The same renderer can be used to render an alpha-map (equivalent to a segmentation map) by setting the colors for each Gaussian to be 1. The Gaussian Splatting model for the object of interest is optimized by minimizing this loss function across multiple views:

$$L = \sum_t (I_t \odot m_t - \Pi(G, C_i))^2$$

where m_t and I_t represents the observed 2D segmentation map and RGB values in frame t respectively.

Projecting Amodal Segmentations: Once we obtain a satisfactory 3D reconstruction of the object, we can generate amodal segmentations for any desired viewpoint. This is done by rendering the entire 3D Gaussian representation back onto the image plane, regardless of occlusions present in the original views. As explained above, we set the Gaussian *colors* to 1 which provides an alpha map using the renderer as

$$\hat{m} = \Pi(G, C_i) \in \mathbb{R}^{H \times W}$$

where $\hat{m}$ is the amodal segmentation map. This map represents the full extent of the object, including parts that may be occluded in the original views. The values in $\hat{m}$ range from 0 to 1, indicating the likelihood of each pixel belonging to the object.

This approach allows us to generate accurate amodal segmentations that account for the full 3D structure of the object, providing a more complete representation

[1]We use 2D Gaussian Splatting [Huang et al. 2024] which is a variation of 3D Gaussian Splatting [Kerbl et al. 2023] that makes it more straightforward to obtain object meshes.

than what is directly observable in any single frame of the video.

5.6.8 Limitations

Our method significantly improves object tracking in egocentric videos, especially under conditions of rapid motion, occlusions, and out-of-sight objects. However, there are limitations, particularly in scenarios where accurate camera poses are difficult to obtain or estimate. Specifically, our approach relies heavily on the assumption that high-quality camera intrinsics and extrinsics are available, as they are essential for accurate 3D lifting of object segments. Hence, performance can degrade in cases with noisy depth maps or challenging conditions like motion blur, poor lighting, or extreme viewpoint changes, as these factors reduce the precision of 3D reconstruction.

Chapter 6

Discussion

6.1 Summary and Impact

This section summarizes the main contributions of the thesis, discusses their broader impact, and concludes with directions for future research extending this work.

6.1.1 Geometry-aware Learning

In Chapter 2, we presented the paper "**A Light Touch Approach to Teaching Transformers Multi-view Geometry**" which explores how transformers can internalise geometric structure without explicit supervision. We introduced an epipolar-aware attention objective that encourages cross-view consistency by softly constraining attention maps along epipolar lines. The model learns viewpoint-invariant object matching without camera pose at inference, outperforming pose-dependent baselines on large-scale retrieval benchmarks. This work showed that geometric reasoning can emerge within attention mechanisms through differentiable, weak supervision, connecting classical multi-view geometry with modern large-scale pretraining.

The paper in Chapter 2 has inspired several follow-up works. [Leroy et al. 2024] introduces MASt3R, which builds upon epipolar geometry principles to ground image matching in 3D space for improved reconstruction. [Kloepfer et al. 2024] extends epipolar supervision concepts for subpixel correspondence estimation in 3D vision applications. [Zhan et al. 2025] leverages multi-view geometric under-

standing to enhance visual-language foundation models for more robust image retrieval across diverse viewpoints.

Limitations. While pose is not needed at inference time, training relies on epipolar geometry; when pseudo-geometry is noisy, the gains reduce. The benefits are confined to the reranking stage and can diminish for highly repetitive textures.

6.1.2 Object-centric 3D Learning

In Chapter 3, we presented the paper "**Contrastive Lift: 3D Object Instance Segmentation by Slow-Fast Contrastive Fusion**" that addresses how 2D segmentations from foundation models can be lifted into 3D, even when the 2D labels are *inconsistent* across views. The proposed Contrastive Lift method fuses multi-view instance masks through slow-fast contrastive alignment to construct coherent 3D neural fields without 3D labels. The resulting representations segment and reconstruct hundreds of objects in cluttered scenes and generalise across real and synthetic datasets, including the Messy Rooms benchmark. This work demonstrated that 2D foundation models, when fused geometrically, provide a scalable route toward object-centric 3D perception.

Messy Rooms dataset. To study the scalability of our approach, we developed a synthetic benchmark of indoor scenes containing up to *several hundred* objects per environment. Each sequence provides multi-view RGB frames and ground-truth segmentations for evaluating multi-object reconstruction and segmentation. The dataset features heavy occlusion and clutter conditions under which many existing 3D instance segmentation methods fail, and serves as a challenging test-bed for such methods.

The work in Chapter 3 has inspired several follow-up works. [R. Zhu et al. 2025] extends our contrastive fusion approach for end-to-end 2D to 3D scene segmentation using Gaussian Splatting representations. [M. Chen et al. 2024] builds upon our multi-view instance segmentation framework for part-level 3D generation and reconstruction. [H. Shen et al. 2025] adapts our slow-fast contrastive learning methodology for consistent segmentation lifting via Gaussian instance tracing. [Pan et al. 2025] develops interactive 2D-to-3D segmentation by linking

2D prompts to our 3D instance representations.

Limitations. The approach assumes reliable multi-view reconstruction; catastrophic 2D failures (e.g., entirely missing object classes) cannot be recovered. We focus on static scenes, as reconstruction remains unreliable in dynamic settings.

6.1.3 Hierarchical and Multimodal Understanding

In Chapter 4, we presented the paper "**N2F2: Hierarchical Scene Understanding with Nested Neural Feature Fields**" which introduces a unified representation for hierarchical and multimodal 3D understanding. N2F2 encodes hierarchies with scenes into a single continuous neural field, assigning subspaces to different semantic levels and aligning them with vision-language embeddings. This allows open-vocabulary querying and text-guided interaction within 3D environments. The model achieves faster inference and improved accuracy compared with prior language-aware neural fields, revealing how geometry and language can coexist within a shared 3D feature space.

The paper in Chapter 4 has inspired several follow-up works. [Cen et al. 2025] extends our nested feature field architecture to handle view-dependent semantics in 3D language Gaussian Splatting. [Sheng et al. 2025] builds upon our hierarchical representation for efficient semantic 3D reconstruction from sparse unposed images. [B. Zhang et al. 2025] adapts our multimodal framework for fine-grained 3D part segmentation with minimal user interaction.

Limitations. Performance depends on the quality of underlying reconstructions. While effective for object- and part-level queries, it can be brittle for queries reflecting multi-step scene-level reasoning.

6.1.4 Dynamic Scene Understanding

In Chapter 5, we presented the paper "**3D-Aware Instance Segmentation and Tracking in Egocentric Videos**" that extends these ideas to dynamic settings through 3D-aware instance segmentation and tracking in egocentric videos. By integrating 2D segmentation, geometric lifting, and 3D association, the model tracks object identities consistently through motion and occlusion, enabling amodal 3D

reconstruction over time. This formulation connects object-centric 3D perception with temporal reasoning, advancing toward persistent, space-time (4D) representations of everyday scenes.

The paper in Chapter 5 has inspired several follow-up works. [Tschernezki et al. 2025] extends our 3D-aware tracking framework by introducing layered motion fusion for lifting motion segmentation to 3D in egocentric videos. [Mur-Labadia et al. 2025] builds upon our geometric lifting and temporal association methods to create dynamic image-video feature fields for enhanced environment understanding in first-person videos.

Limitations. The pipeline may inherit errors from the depth/pose used for 3D lifting; rapid egomotion, motion blur can still cause identity switches. We assume largely rigid objects which may limit robustness in highly deformable scenarios.

6.2 Future Work

The methods developed in this thesis suggest several directions for future exploration.

Compositional Scene Generation

An immediate next step is to extend object-centric representations toward generative modelling. Instead of reconstructing existing scenes, a model could compose new scenes by arranging individually generated object-level fields under learned physical or spatial constraints. Such compositional generation would unify recognition and synthesis, allowing controllable creation of plausible 3D environments consistent with geometric and semantic priors.

World-model Learning

A longer-term direction is to develop world models that learn the dynamics of 3D environments over time. These systems would not only reconstruct observed scenes but also predict their evolution by linking perception, memory, and imagination. By integrating neural field representations with recurrent or predictive architectures, future models could simulate how objects and agents interact, forming the

basis for forecasting future world states in real-world settings.

Interactive and Embodied Applications

Finally, coupling 3D understanding with action and control opens pathways for robotics and augmented reality. Object-centric neural fields can serve as perceptual front-ends for manipulation, navigation, and interaction, enabling agents to perceive, anticipate, and reshape their environment.

6.3 Conclusion

This thesis advanced object-centric 3D scene understanding from video through four complementary contributions: geometry-aware feature learning, unsupervised 3D reconstruction, hierarchical multimodal reasoning, and dynamic scene tracking. Collectively, they show that geometry, semantics, and motion can be unified within continuous neural field representations guided by large-scale visual and language priors. The resulting framework moves closer to the long-standing goal of transforming passive video into an active model of the world – a foundation for future systems that can perceive, reason, and generate within the same coherent 3D space.

References

Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky (2022). "BoT-SORT: Robust associations multi-pedestrian tracking". In: *arXiv preprint arXiv:2206.14651*.

Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. (2021). "Xcit: Cross-covariance image transformers". In: *Advances in neural information processing systems*.

Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic (2016). "NetVLAD: CNN architecture for weakly supervised place recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Relja Arandjelović and Andrew Zisserman (2011). "Smooth Object Retrieval using a Bag of Boundaries". In: *ICCV*.

Relja Arandjelović and Andrew Zisserman (2012). "Three things everyone should know to improve object retrieval". In: *CVPR*.

Ali Athar, Jonathon Luiten, Paul Voigtlaender, Tarasha Khurana, Achal Dave, Bastian Leibe, and Deva Ramanan (2023). "Burst: A benchmark for unifying object recognition, segmentation and tracking in video". In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*.

Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky (2014). "Neural codes for image retrieval". In: *European conference on computer vision*. Springer.

Yutong Bai, Angtian Wang, Adam Kortylewski, and Alan Yuille (2023). "CoKe: Contrastive Learning for Robust Keypoint Detection". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.

Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas (2020). "MAGSAC++, a fast, reliable and accurate robust estimator". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan (2021). "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani (2021). "Is space-time attention all you need for video understanding?" In: *ICML*.

Yash Bhalgat, Iro Laina, João F Henriques, Andrea Vedaldi, and Andrew Zisserman (2023). "Contrastive Lift: 3D Object Instance Segmentation by Slow-Fast Contrastive Fusion". In: *Advances in Neural Information Processing Systems*.

Yash Bhalgat, Iro Laina, João F Henriques, Andrew Zisserman, and Andrea Vedaldi (2024). "N2F2: Hierarchical Scene Understanding with Nested Neural Feature Fields". In: *arXiv preprint arXiv:2403.10997*.

Malik Boudiaf, Jérôme Rony, Imtiaz Masud Ziko, Eric Granger, Marco Pedersoli, Pablo Piantanida, and Ismail Ben Ayed (2020). "Metric learning: cross-entropy vs. pairwise losses". In: *arXiv preprint arXiv:2003.08983*.

Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother (2017). "Dsac-differentiable ransac for camera localization". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Christopher P. Burgess, Löic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matthew Botvinick, and Alexander Lerchner (2019). "MONet: Unsupervised Scene Decomposition and Representation". In: *arXiv.cs*.

Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool (2019). "The 2019 davis challenge on vos: Unsupervised multi-object segmentation". In: *arXiv preprint arXiv:1905.00737*.

Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff (2019). "Deep metric learning to rank". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Bingyi Cao, Andre Araujo, and Jack Sim (2020). "Unifying deep local and global features for image search". In: *European Conference on Computer Vision*. Springer.

Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani (2023). "Observation-centric sort: Rethinking sort for robust multi-object tracking". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko (2020). "End-to-end object detection with transformers". In: *European conference on computer vision*. Springer.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin (2020). "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments". In: *Proc. NeurIPS*.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin (2021). "Emerging Properties in Self-Supervised Vision Transformers". In: *Proc. ICCV*.

Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian (2023). "Segment any 3d gaussians". In: *arXiv preprint arXiv:2312.00860*.

Jiazhong Cen, Xudong Zhou, Jiemin Fang, Changsong Wen, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian (2025). "Tackling View-Dependent Semantics in 3D Language Gaussian Splatting". In: *arXiv.org*.

Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, et al. (2024). "Segment anything in 3d with nerfs". In: *Advances in Neural Information Processing Systems*.

Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein (2022). "Efficient Geometry-aware 3D Generative Adversarial Networks". In: *Proc. CVPR*.

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su (2022). "TensoRF: Tensorial Radiance Fields". In: *European Conference on Computer Vision (ECCV)*.

Haoran Chen, Kenneth Blomqvist, Francesco Milano, and Roland Siegwart (2024). "Panoptic vision-language feature fields". In: *IEEE Robotics and Automation Letters*.

Minghao Chen, Roman Shapovalov, Iro Laina, Tom Monnier, Jianyuan Wang, David Novotný, and Andrea Vedaldi (2024). "PartGen: Part-level 3D Generation and Reconstruction with Multi-View Diffusion Models". In: *Computer Vision and Pattern Recognition*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton (2020). "A Simple Framework for Contrastive Learning of Visual Representations". In: *Proc. ICML*.

Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin (2023). "GaussianEditor: Swift and Controllable 3D Editing with Gaussian Splatting". In: *arXiv.cs*.

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.

Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee (2023). "Tracking anything with decoupled video segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Ho Kei Cheng and Alexander G Schwing (2022). "Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model". In: *European Conference on Computer Vision*. Springer.

Lele Cheng, Xiangzeng Zhou, Liming Zhao, Dangwei Li, Hong Shang, Yun Zheng, Pan Pan, and Yinghui Xu. (Aug. 2020). "Weakly Supervised Learning with Side Information for Noisy Labeled Images". In: *The European Conference on Computer Vision (ECCV)*.

Sumit Chopra, Raia Hadsell, and Yann LeCun (2005). "Learning a similarity metric discriminatively, with application to face verification". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. IEEE.

Anwesa Choudhuri, Girish Chowdhary, and Alexander G Schwing (2021). "Assignment-space-based multi-object tracking and segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Anwesa Choudhuri, Girish Chowdhary, and Alexander G Schwing (2023). "Context-aware relative object queries to unify video instance and panoptic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman (2007). "Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval". In: *Proc. ICCV*.

Ondřej Chum, Andrej Mikulik, Michal Perdoch, and Jiří Matas (2011). "Total recall II: Query expansion revisited". In: *CVPR 2011*. IEEE.

Dorin Comaniciu and Peter Meer (2002). "Mean shift: A robust approach toward feature space analysis". In: *IEEE Transactions on pattern analysis and machine intelligence*.

Blender Online Community (2018). *Blender - a 3D modelling and rendering package*. Blender Foundation. Stichting Blender Foundation, Amsterdam. URL: http://www.blender.org.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner (2017). "Scannet: Richly-annotated 3d reconstructions of indoor scenes". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray (2021). "The EPIC-KITCHENS Dataset: Collection, Challenges and Baselines". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Bert De Brabandere, Davy Neven, and Luc Van Gool (2017). "Semantic instance segmentation with a discriminative loss function". In: *arXiv preprint arXiv:1708.02551*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee.

Karan Desai, Maximilian Nickel, Tanmay Rajpurohit, Justin Johnson, and Shanmukha Ramakrishna Vedantam (2023). "Hyperbolic image-text representations". In: *International Conference on Machine Learning*. PMLR.

Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich (2017). "Toward geometric deep slam". In: *arXiv preprint arXiv:1707.07410*.

Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich (2018). "Superpoint: Self-supervised interest point detection and description". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*.

Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi (2023). "PLA: Language-Driven Open-Vocabulary 3D Scene Understanding". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas 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*.

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*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. (2020). "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929*.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke (2022). "Google scanned objects: A high-quality dataset of 3d scanned household items". In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE.

Francis Engelmann, Fabian Manhardt, Michael Niemeyer, Keisuke Tateno, and Federico Tombari (2023). "OpenNerf: Open Set 3D Neural Scene Segmentation with Pixel-Wise Features and Rendered Novel Views". In: *The Twelfth International Conference on Learning Representations*.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. (1996). "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *kdd*.

Jose M Facil, Benjamin Ummenhofer, Huizhong Zhou, Luis Montesano, Thomas Brox, and Javier Civera (2019). "CAM-Convs: Camera-aware multi-scale convolutions for single-view depth". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer (2021). "Multiscale vision transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Zhiwen Fan, Peihao Wang, Yifan Jiang, Xinyu Gong, Dejia Xu, and Zhangyang Wang (2023). "NeRF-SOS: Any-View Self-supervised Object Segmentation on Complex Scenes". In: *The Eleventh International Conference on Learning Representations*.

Jiemin Fang, Junjie Wang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian (2023). "GaussianEditor: Editing 3D Gaussians Delicately with Text Instructions". In: *arXiv.cs*.

Alireza Fathi, Zbigniew Wojna, Vivek Rathod, Peng Wang, Hyun Oh Song, Sergio Guadarrama, and Kevin P Murphy (2017). "Semantic instance segmentation via deep metric learning". In: *arXiv preprint arXiv:1703.10277*.

William Fedus, Barret Zoph, and Noam Shazeer (2021). *Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity*.

Martin A Fischler and Robert C Bolles (1981). “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”. In: *Communications of the ACM*.

Sara Fridovich-Keil, Giacomo Meanti, Frederik Warburg, Benjamin Recht, and Angjoo Kanazawa (2023). “K-Planes: Explicit Radiance Fields in Space, Time, and Appearance”. In: *arXiv.cs*.

Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao (2022). “Panoptic NeRF: 3D-to-2D Label Transfer for Panoptic Urban Scene Segmentation”. In: *arXiv.cs*.

Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien P. C. Valentin (2021). “FastNeRF: High-Fidelity Neural Rendering at 200FPS”. In: *CoRR*.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann (2020). “Shortcut learning in deep neural networks”. In: *Nature Machine Intelligence*.

Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman (2019). “Video action transformer network”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Rahul Goel, Dhawal Sirikonda, Saurabh Saini, and PJ Narayanan (2023). “Interactive segmentation of radiance fields”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus (2016). “Deep image retrieval: Learning global representations for image search”. In: *European conference on computer vision*. Springer.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. (2022). “Ego4d: Around the world in 3,000 hours of egocentric video”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

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.

Klaus Greff, Raphael Lopez Kaufmann, 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 36th International Conference on Machine Learning*. PMLR.

Qiao Gu, Zhaoyang Lv, Duncan Frost, Simon Green, Julian Straub, and Chris Sweeney (2024). “EgoLifter: Open-world 3D Segmentation for Egocentric Perception”. In: *arXiv preprint arXiv:2403.18118*.

Agrim Gupta, Piotr Dollar, and Ross Girshick (2019). “Lvis: A dataset for large vocabulary instance segmentation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Huy Ha and Shuran Song (2022). “Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models”. In: *6th Annual Conference on Robot Learning*.

Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang (2021). “Transformer in transformer”. In: *Advances in Neural Information Processing Systems*.

Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki (2022). “Particle video revisited: Tracking through occlusions using point trajectories”. In: *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.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer.

Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer (2021). “Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick (2019). “Momentum Contrast for Unsupervised Visual Representation Learning”. In: *arXiv.cs*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). “Deep Residual Learning for Image Recognition”. In: *CVPR*. IEEE Computer Society.

Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu (2020). "Epipolar transformers". In: *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*.

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: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*. Springer.

Alexander Hermans, Georgios Floros, and Bastian Leibe (2014). "Dense 3d semantic mapping of indoor scenes from rgb-d images". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.

Xu Hu, Yuxi Wang, Lue Fan, Junsong Fan, Junran Peng, Zhen Lei, Qing Li, and Zhaoxiang Zhang (2024). "Semantic Anything in 3D Gaussians". In: *arXiv preprint arXiv:2401.17857*.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao (2024). "2D Gaussian Splatting for Geometrically Accurate Radiance Fields". In: *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira (2021). "Perceiver: General perception with iterative attention". In: *International conference on machine learning*. PMLR.

Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, et al. (2023). "Conceptfusion: Open-set multimodal 3d mapping". In: *arXiv preprint arXiv:2302.07241*.

H. Jégou, M. Douze, and C. Schmid (2008). "Hamming embedding and weak geometric consistency for large scale image search". In: *ECCV*.

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: *International Conference on Machine Learning*. PMLR.

Yushi Jing and Shumeet Baluja (2008). "Pagerank for product image search". In: *Proceedings of the 17th international conference on World Wide Web*.

Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei (2015). "Image retrieval using scene graphs". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Amita Kamath, Jack Hessel, and Kai-Wei Chang (2023). "Text encoders bottleneck compositionality in contrastive vision-language models". In: *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht (2023). "Cotracker: It is better to track together". In: *arXiv preprint arXiv:2307.07635*.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis (July 2023). "3D Gaussian Splatting for Real-Time Radiance Field Rendering". In: *ACM Transactions on Graphics*. URL: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik (2023). "LERF: Language Embedded Radiance Fields". In: *arXiv preprint arXiv:2303.09553*.

Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa (2024a). "GARField: Group Anything with Radiance Fields". In: *arXiv.cs*.

Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa (2024b). "Garfield: Group anything with radiance fields". In: *CVPR*.

Alexander Kirillov, Kaiming He, Ross B. Girshick, Carsten Rother, and Piotr Dollár (2019). "Panoptic Segmentation". In: *Proc. CVPR*.

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: *ICCV*.

Dominik A. Kloepfer, João F. Henriques, and Dylan Campbell (2024). "SCENES: Subpixel Correspondence EstimationWith Epipolar Supervision". In: *International Conference on 3D Vision*.

Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann (2022a). "Decomposing NeRF for Editing via Feature Field Distillation". In: *arXiv.cs*.

Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann (2022b). "Decomposing nerf for editing via feature field distillation". In: *NeurIPS*.

Shu Kong and Charless C Fowlkes (2018). "Recurrent pixel embedding for instance grouping". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Josip Krapac, Moray Allan, Jakob Verbeek, and Frédéric Jurie (2010). "Improving web image search results using query-relative classifiers". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser (2022). "Panoptic Neural Fields: A Semantic Object-Aware Neural Scene Representation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Abhijit Kundu, Yin Li, Frank Dellaert, Fuxin Li, and James M Rehg (2014). "Joint semantic segmentation and 3d reconstruction from monocular video". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*. Springer.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, et al. (2022). "Matryoshka representation learning". In: *Advances in Neural Information Processing Systems*.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud (2024). "Grounding Image Matching in 3D with MASt3R". In: *European Conference on Computer Vision*.

Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl (2022). "Language-driven Semantic Segmentation". In: *ICLR*. URL: https://openreview.net/forum?id=RriDjddCLN.

Siyuan Li, Lei Ke, Martin Danelljan, Luigi Piccinelli, Mattia Segu, Luc Van Gool, and Fisher Yu (2024). "Matching Anything By Segmenting Anything". In: *CVPR*.

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: *CVPR*.

Kevin Lin, Lijuan Wang, and Zicheng Liu (2021). "End-to-end human pose and mesh reconstruction with transformers". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

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: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer.

Zhiqiu Lin, Xinyue Chen, Deepak Pathak, Pengchuan Zhang, and Deva Ramanan (2023). "VisualGPTScore: Visio-Linguistic Reasoning with Multimodal Generative Pre-Training Scores". In: *arXiv preprint arXiv:2306.01879*.

Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, Abdulmotaleb El Saddik, Christian Theobalt, Eric Xing, and Shijian Lu (2023). "Weakly Supervised 3D Open-vocabulary Segmentation". In: *NeurIPS*.

Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt (2020). "Neural sparse voxel fields". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. (2023). "Grounding dino: Marrying dino with grounded pre-training for open-set object detection". In: *arXiv preprint arXiv:2303.05499*.

Yichen Liu, Benran Hu, Junkai Huang, Yu-Wing Tai, and Chi-Keung Tang (2023). "Instance neural radiance field". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

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*.

Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Peng Yin, Wen Chen, Hesheng Wang, Haoang Li, and Yun-Hui Liu (2019). "Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis". 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*.

H Christopher Longuet-Higgins (1981). "A computer algorithm for reconstructing a scene from two projections". In: *Nature*.

David G Lowe (1999). "Object recognition from local scale-invariant features". In: *Proceedings of the seventh IEEE international conference on computer vision*. Ieee.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee (2019). "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks". In: *Advances in neural information processing systems*.

Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe (2021). "Hota: A higher order metric for evaluating multi-object tracking". In: *International journal of computer vision*.

Jin Ma, Shanmin Pang, Bo Yang, Jihua Zhu, and Yaochen Li (2020). "Spatial-content image search in complex scenes". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.

Lingni Ma, Jörg Stückler, Christian Kerl, and Daniel Cremers (2017). "Multi-view deep learning for consistent semantic mapping with RGB-D cameras". In: *Proc.IROS*.

Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna (2023). "CREPE: Can Vision-Language Foundation Models Reason Compositionally?" In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Gerard Maggiolino, Adnan Ahmad, Jinkun Cao, and Kris Kitani (2023). "Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification". In: *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. USA: Cambridge University Press.

Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth (2021). "NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections". In: *Proc. CVPR*.

Ruben Mascaro, Lucas Teixeira, and Margarita Chli (2021). "Diffuser: Multi-View 2D-to-3D Label Diffusion for Semantic Scene Segmentation". In: *Proc. ICRA*.

Neil Houlsby Matthias Minderer Alexey Gritsenko (2023). "Scaling Open-Vocabulary Object Detection". In: *NeurIPS*.

John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger (2017). "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks". In: *2017 IEEE International Conference on Robotics and automation (ICRA)*. IEEE.

Leland McInnes, John Healy, and Steve Astels (2017). "hdbscan: Hierarchical density based clustering." In: *J. Open Source Softw.*

Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer (2022). "Trackformer: Multi-object tracking with transformers". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool (2005). "A Comparison of Affine Region Detectors". In: *IJCV*.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng (2020). "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". In: *ECCV*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm.

Ashkan Mirzaei, Yash Kant, Jonathan Kelly, and Igor Gilitschenski (2022). "LaTeRF: Label and text driven object radiance fields". In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*. Springer.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller (July 2022). "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding". In: *ACM Trans. Graph.* URL: https://doi.org/10.1145/3528223.3530127.

Lorenzo Mur-Labadia, Jose J. Guerrero, and Ruben Martinez-Cantin (2025). "DIV-FF: Dynamic Image-Video Feature Fields For Environment Understanding in Egocentric Videos". In: *Computer Vision and Pattern Recognition*.

Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji (2019). "Panopticfusion: Online volumetric semantic mapping at the level of stuff and things". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann (2021). "Video transformer network". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Alejandro Newell, Zhiao Huang, and Jia Deng (2017). "Associative embedding: End-to-end learning for joint detection and grouping". In: *Advances in neural information processing systems*.

Zaiqing Nie, Yunxiao Ma, Shuming Shi, Ji-Rong Wen, and Wei-Ying Ma (2007). "Web object retrieval". In: *Proceedings of the 16th international conference on World Wide Web*.

Michael Niemeyer and Andreas Geiger (2020). "GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields". In: *https://arxiv.org/abs/2011.12100*.

David Nistér (2004). "An efficient solution to the five-point relative pose problem". In: *IEEE transactions on pattern analysis and machine intelligence*.

Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han (2017). "Large-scale image retrieval with attentive deep local features". In: *Proceedings of the IEEE international conference on computer vision*.

Alaaeldin El-Nouby, Natalia Neverova, Ivan Laptev, and Hervé Jégou (2021). "Training vision transformers for image retrieval". In: *arXiv preprint arXiv:2102.05644*.

David Novotny, Samuel Albanie, Diane Larlus, and Andrea Vedaldi (2018). "Semi-convolutional operators for instance segmentation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Amadeus Oertel, Titus Cieslewski, and Davide Scaramuzza (2020). "Augmenting visual place recognition with structural cues". In: *IEEE Robotics and Automation Letters*.

Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim (2019). "Video object segmentation using space-time memory networks". In: *Proceedings of the IEEE/CVF international conference on computer vision*.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese (2016). "Deep metric learning via lifted structured feature embedding". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski (2023). *DINOv2: Learning Robust Visual Features without Supervision*.

Honghan Pan, Bangzhen Liu, Xuemiao Xu, Chenxi Zheng, Yongwei Nie, and Shengfeng He (2025). "Gaussian Prompter: Linking 2D Prompts for 3D Gaussian Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan Goldman, Steven Seitz, and Ricardo Martin-Brualla (2020). "Deformable Neural Radiance Fields". In: *https://arxiv.org/abs/2011.12948*.

Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla (2021). "Nerfies: Deformable Neural Radiance Fields". In: *ICCV*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. (2019). "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems*.

Nilay Patel and Jeffrey Flanigan (2022). "Forming trees with treeformers". In: *arXiv preprint arXiv:2207.06960*.

Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. (2023). "Openscene: 3d scene understanding with open vocabularies". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander 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*.

James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman (2007). "Object Retrieval with Large Vocabularies and Fast Spatial Matching". In: *Proc. CVPR*.

Chiara Plizzari, Marco Cannici, and Matteo Matteucci (2021). "Spatial temporal transformer network for skeleton-based action recognition". In: *International Conference on Pattern Recognition*. Springer.

Chiara Plizzari, Shubham Goel, Toby Perrett, Jacob Chalk, Angjoo Kanazawa, and Dima Damen (2024). "Spatial Cognition from Egocentric Video: Out of Sight, Not Out of Mind". In: *ArXiv*.

Vignesh Prasad, Dipanjan Das, and Brojeshwar Bhowmick (2018). "Epipolar geometry based learning of multi-view depth and ego-motion from monocular sequences". In: *Proceedings of the 11th Indian Conference on Computer Vision, Graphics and Image Processing*.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer (2020). "D-NeRF: Neural Radiance Fields for Dynamic Scenes". In: *arXiv.cs*.

Siyuan Qiao, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen (2021). "Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister (2023). "LangSplat: 3D Language Gaussian Splatting". In: *arXiv preprint arXiv:2312.16084*.

Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie (2007). "Objects in context". In: *2007 IEEE 11th International Conference on Computer Vision*. IEEE.

Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum (2018). "Revisiting oxford and paris: Large-scale image retrieval benchmarking". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

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: *International Conference on Machine Learning*. PMLR.

Jathushan Rajasegaran, Georgios Pavlakos, Angjoo Kanazawa, and Jitendra Malik (June 2022). "Tracking People by Predicting 3D Appearance, Location and Pose". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Frano Rajič, Lei Ke, Yu-Wing Tai, Chi-Keung Tang, Martin Danelljan, and Fisher Yu (2023). "Segment anything meets point tracking". In: *arXiv preprint arXiv:2307.01197*.

Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger (2021). "KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs". In: *CoRR*.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny (2021). "Common Objects in 3D: Large-Scale Learning and Evaluation of Real-life 3D Category Reconstruction". In: *International Conference on Computer Vision*.

Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger (2019). "R2D2: repeatable and reliable detector and descriptor". In: *arXiv preprint arXiv:1906.06195*.

Helge Rhodin, Mathieu Salzmann, and Pascal Fua (2018). "Unsupervised geometry-aware representation for 3d human pose estimation". In: *Proceedings of the European conference on computer vision (ECCV)*.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind (2021). "Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Chris Rockwell, Justin Johnson, and David F Fouhey (2022). "The 8-Point Algorithm as an Inductive Bias for Relative Pose Prediction by ViTs". In: *arXiv preprint arXiv:2208.08988*.

Edward Rosten and Tom Drummond (2006). "Machine learning for high-speed corner detection". In: *European conference on computer vision*. Springer.

Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Bjorn Ommer, and Joseph Paul Cohen (2020). "Revisiting training strategies and generalization

performance in deep metric learning". In: *International Conference on Machine Learning*. PMLR.

Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski (2011). "ORB: An efficient alternative to SIFT or SURF". In: *2011 International conference on computer vision*. Ieee.

John W Santrock (2002). *A topical approach to life-span development*. McGraw Hill.

Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa (2022). "Plenoxels: Radiance Fields without Neural Networks". In: *CVPR*.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich (2020). "Superglue: Learning feature matching with graph neural networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Johannes L Schonberger and Jan-Michael Frahm (2016). "Structure-from-motion revisited". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Johannes Lutz Schönberger and Jan-Michael Frahm (2016). "Structure-from-Motion Revisited". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm (2016). "Pixelwise View Selection for Unstructured Multi-View Stereo". In: *European Conference on Computer Vision (ECCV)*.

Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger (2020). "Graf: Generative radiance fields for 3D-aware image synthesis". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Prafull Sharma, Ayush Tewari, Yilun Du, Sergey Zakharov, Rares Andrei Ambrus, Adrien Gaidon, William T. Freeman, Fredo Durand, Joshua B. Tenenbaum, and Vincent Sitzmann (2023). "Neural Groundplans: Persistent Neural Scene Representations from a Single Image". In: *The Eleventh International Conference on Learning Representations*.

Hongyu Shen, Junfeng Ni, Yixin Chen, Weishuo Li, Mingtao Pei, and Siyuan Huang (2025). "Trace3D: Consistent Segmentation Lifting via Gaussian Instance Tracing". In: *arXiv.org*.

Yu Sheng, Jiajun Deng, Xinran Zhang, Yu Zhang, Bei Hua, Yanyong Zhang, and Jianmin Ji (2025). "SpatialSplat: Efficient Semantic 3D from Sparse Unposed Images". In: *arXiv.org*.

Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder (June 2023a). "Panoptic Lifting for 3D Scene Understanding With Neural Fields". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder (2023b). "Panoptic Lifting for 3D Scene Understanding with Neural Fields". In: *Proc. CVPR*.

Karen Simonyan and Andrew Zisserman (2015). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations*.

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein (2019). "Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations". In: *NeurIPS*. Ed. by Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. (2019). "The Replica dataset: A digital replica of indoor spaces". In: *arXiv preprint arXiv:1906.05797*.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai (2020). "VL-BERT: Pre-training of Generic Visual-Linguistic Representations". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=SygXPaEYvH.

Cheng Sun, Min Sun, and Hwann-Tzong Chen (2022). "Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction". In: *Proc. CVPR*.

Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou (2021). "LoFTR: Detector-free local feature matching with transformers". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Niko Sünderhauf, Trung T Pham, Yasir Latif, Michael Milford, and Ian Reid (2017). "Meaningful maps with object-oriented semantic mapping". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii (2018). "InLoc: Indoor visual localization with dense matching and view synthesis". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Fuwen Tan, Jiangbo Yuan, and Vicente Ordonez (2021). “Instance-level image retrieval using reranking transformers”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa (2023). “Nerfstudio: A Modular Framework for Neural Radiance Field Development”. In: *ACM SIGGRAPH 2023 Conference Proceedings*. SIGGRAPH ’23.

Hao Tang, Kevin J Liang, Kristen Grauman, Matt Feiszli, and Weiyao Wang (2024). “Egotracks: A long-term egocentric visual object tracking dataset”. In: *Advances in Neural Information Processing Systems*.

Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng (2023). “DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation”. In: *arXiv*.

Antti Tarvainen and Harri Valpola (2017). “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results”. In: *Proc. NeurIPS*.

Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab (2017). “Cnn-slam: Real-time dense monocular slam with learned depth prediction”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross (2022). “Winoground: Probing vision and language models for visio-linguistic compositionality”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Giorgos Tolias and Hervé Jégou (2014). “Visual query expansion with or without geometry: refining local descriptors by feature aggregation”. In: *Pattern recognition*.

Giorgos Tolias, Ronan Sicre, and Hervé Jégou (2015). “Particular object retrieval with integral max-pooling of CNN activations”. In: *arXiv preprint arXiv:1511.05879*.

Akihiko Torii, Josef Sivic, Tomas Pajdla, and Masatoshi Okutomi (2013). “Visual place recognition with repetitive structures”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Michael Tschannen, Josip Djolonga, Paul K. Rubenstein, Sylvain Gelly, and Mario Lucic (2020). “On Mutual Information Maximization for Representation Learning”. In: *Proc. ICLR*.

Vadim Tschernezki, Ahmad Darkhalil, Zhifan Zhu, David Fouhey, Iro Larina, Diane Larlus, Dima Damen, and Andrea Vedaldi (2023). “EPIC Fields: Marrying

3D Geometry and Video Understanding". In: *Proceedings of the Neural Information Processing Systems (NeurIPS)*.

Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi (2022). "Neural Feature Fusion Fields: 3D Distillation of Self-Supervised 2D Image Representation". In: *Proceedings of the International Conference on 3D Vision (3DV)*.

Vadim Tschernezki, Diane Larlus, and Andrea Vedaldi (2021). "NeuralDiff: Segmenting 3D objects that move in egocentric videos". In: *Proceedings of the International Conference on 3D Vision (3DV)*.

Vadim Tschernezki, Diane Larlus, Andrea Vedaldi, and Iro Laina (2025). "Layered Motion Fusion: Lifting Motion Segmentation to 3D in Egocentric Videos". In: *Computer Vision and Pattern Recognition*.

Shubham Tulsiani, Alexei A Efros, and Jitendra Malik (2018). "Multi-view consistency as supervisory signal for learning shape and pose prediction". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik (2017). "Multi-view supervision for single-view reconstruction via differentiable ray consistency". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Nikolai Ufer, Max Simon, Sabine Lang, and Björn Ommer (2021). "Large-scale interactive retrieval in art collections using multi-style feature aggregation". In: *PloS one*.

Mikaela Angelina Uy and Gim Hee Lee (2018). "Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals (2019). "Representation Learning with Contrastive Predictive Coding". In: *Proc. NeurIPS*.

Reinier H Van Leuken, Lluis Garcia, Ximena Olivares, and Roelof van Zwol (2009). "Visual diversification of image search results". In: *Proceedings of the 18th international conference on World wide web*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *Advances in neural information processing systems*.

Vibhav Vineet, Ondrej Miksik, Morten Lidegaard, Matthias Nießner, Stuart Golodetz, Victor A Prisacariu, Olaf Kähler, David W Murray, Shahram Izadi, Patrick Pérez, et al. (2015). "Incremental dense semantic stereo fusion for large-scale semantic

scene reconstruction". In: *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE.

Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen (2019). "Feelvos: Fast end-to-end embedding learning for video object segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Suhani Vora, Noha Radwan, Klaus Greff, Henning Meyer, Kyle Genova, Mehdi S. M. Sajjadi, Etienne Pot, Andrea Tagliasacchi, and Daniel Duckworth (2021). "NeSF: Neural Semantic Fields for Generalizable Semantic Segmentation of 3D Scenes". In: *arXiv.cs*.

Bing WANG, Lu Chen, and Bo Yang (2023). "DM-NeRF: 3D Scene Geometry Decomposition and Manipulation from 2D Images". In: *The Eleventh International Conference on Learning Representations*.

Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R Scott (2020). "Cross-batch memory for embedding learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia (2021). "End-to-end video instance segmentation with transformers". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

T. Weyand, A. Araujo, B. Cao, and J. Sim (2020). "Google Landmarks Dataset v2 - A Large-Scale Benchmark for Instance-Level Recognition and Retrieval". In: *Proc. CVPR*.

Olivia Wiles, Sebastien Ehrhardt, and Andrew Zisserman (2021). "Co-Attention for Conditioned Image Matching". In: *CVPR*.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang (2023). "4D Gaussian Splatting for Real-Time Dynamic Scene Rendering". In: *arXiv*.

Junfeng Wu, Yi Jiang, Song Bai, Wenqing Zhang, and Xiang Bai (2022). "Seqformer: Sequential transformer for video instance segmentation". In: *European Conference on Computer Vision*. Springer.

Christopher Xie, Keunhong Park, Ricardo Martin-Brualla, and Matthew Brown (2021). "Fig-nerf: Figure-ground neural radiance fields for 3d object category modelling". In: *2021 International Conference on 3D Vision (3DV)*. IEEE.

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: *CVPR*.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao (2024). “Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data”. In: *CVPR*.

Linjie Yang, Yuchen Fan, and Ning Xu (2019). “Video instance segmentation”. In: *Proceedings of the IEEE/CVF international conference on computer vision*.

Zongxin Yang, Yunchao Wei, and Yi Yang (2021). “Associating objects with transformers for video object segmentation”. In: *Advances in Neural Information Processing Systems*.

Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan (2018). “MVSNet: Depth Inference for Unstructured Multi-view Stereo”. In: *ECCV*. Ed. by Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss. Lecture Notes in Computer Science.

Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke (2024). “Gaussian Grouping: Segment and Edit Anything in 3D Scenes”. In: *ECCV*.

Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua (2016). “Lift: Learned invariant feature transform”. In: *European conference on computer vision*. Springer.

Wang Yifan, Carl Doersch, Relja Arandjelović, João Carreira, and Andrew Zisserman (June 2022). “Input-Level Inductive Biases for 3D Reconstruction”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang (2024). “Omniseg3d: Omniversal 3d segmentation via hierarchical contrastive learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa (2021). “PlenOctrees for Real-time Rendering of Neural Radiance Fields”. In: *arXiv*.

Frank Yu, Mathieu Salzmann, Pascal Fua, and Helge Rhodin (2021). “PCLs: Geometry-aware neural reconstruction of 3D pose with perspective crop layers”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Hong-Xing Yu, Leonidas Guibas, and Jiajun Wu (2022). “Unsupervised Discovery of Object Radiance Fields”. In: *International Conference on Learning Representations*.

Qihang Yu, Huiyu Wang, Dahun Kim, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen (2022a). "Cmt-deeplab: Clustering mask transformers for panoptic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen (2022b). "k-means Mask Transformer". In: *European Conference on Computer Vision*. Springer.

Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan (2021). "Tokens-to-token vit: Training vision transformers from scratch on imagenet". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou (2022). "When and Why Vision-Language Models Behave like Bags-Of-Words, and What to Do About It?" In: *The Eleventh International Conference on Learning Representations*.

Greg Zaal, Rob Tuytel, Rico Cilliers, James Ray Cock, Andreas Mischok, Sergej Majboroda, Dimitrios Savva, and Jurita Burger (2021). *Polyhaven: a curated public asset library for visual effects artists and game designers*. https://polyhaven.com/hdris.

Jesus Zarzar, Sara Rojas, Silvio Giancola, and Bernard Ghanem (2022). "SegNeRF: 3D Part Segmentation with Neural Radiance Fields". In: *arXiv preprint arXiv:2211.11215*.

Guanqi Zhan, Yuanpei Liu, Kai Han, Weidi Xie, and Andrew Zisserman (2025). "ELIP: Enhanced Visual-Language Foundation Models for Image Retrieval". In: *arXiv.org*.

Bojun Zhang, Hangjian Ye, Hao Zheng, Jianzheng Huang, Zhengyu Lin, Zhenhong Guo, and Feng Zheng (2025). *PinPoint3D: Fine-Grained 3D Part Segmentation from a Few Clicks*.

Junbo Zhang, Runpei Dong, and Kaisheng Ma (2023). "Clip-fo3d: Learning free open-world 3d scene representations from 2d dense clip". In: *arXiv preprint arXiv:2303.04748*.

Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang (2022). "ByteTrack: Multi-Object Tracking by Associating Every Detection Box". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun (2021). "Point transformer". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison (2021a). "In-place scene labelling and understanding with implicit scene representation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison (2021b). "In-Place Scene Labelling and Understanding with Implicit Scene Representation". In: *Proc. ICCV*.

Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi (2023). "Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields". In: *arXiv preprint arXiv:2312.03203*.

Tianfei Zhou, Fatih Porikli, David J Crandall, Luc Van Gool, and Wenguan Wang (2022). "A survey on deep learning technique for video segmentation". In: *IEEE transactions on pattern analysis and machine intelligence*.

Wengang Zhou, Yijuan Lu, Houqiang Li, Yibing Song, and Qi Tian (2010). "Spatial coding for large scale partial-duplicate web image search". In: *Proceedings of the 18th ACM international conference on Multimedia*.

Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra (2022). "Detecting Twenty-thousand Classes using Image-level Supervision". In: *ECCV*.

Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl (2020). "Tracking objects as points". In: *European conference on computer vision*. Springer.

Runsong Zhu, Shi Qiu, Zhengzhe Liu, Ka-Hei Hui, Qianyi Wu, Pheng-Ann Heng, and Chi-Wing Fu (2025). "Rethinking End-to-End 2D to 3D Scene Segmentation in Gaussian Splatting". In: *Computer Vision and Pattern Recognition*.

Xingxing Zuo, Pouya Samangouei, Yunwen Zhou, Yan Di, and Mingyang Li (2024). "Fmgs: Foundation model embedded 3d gaussian splatting for holistic 3d scene understanding". In: *arXiv preprint arXiv:2401.01970*.

Appendix A

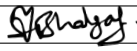
Statement of Authorship

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 "A Light Touch Approach to Teaching Transformers Multi-view Geometry" in Chapter 2.

Paper title	A Light Touch Approach to Teaching Transformers Multi-view Geometry
Authors	**Yash Bhalgat**, João F. Henriques, Andrew Zisserman
Publication status	Published
Publication details	Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

Student Confirmation

Student name	Yash Bhalgat	
Contribution to the paper	First-author contribution: • conception of research ideas • design and implementation of models • writing and presentation of the paper	
Signature and Date		Nov 4th 2025

Supervisor Confirmation

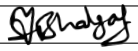
By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Dr. João F. Henriques	
Supervisor comments		
Signature and Date		5/11/2025

Statement of Authorship for the paper "Contrastive Lift: 3D Object Instance Segmentation by Slow-Fast Contrastive Fusion" in Chapter 3.

Paper title	Contrastive Lift: 3D Object Instance Segmentation by Slow-Fast Contrastive Fusion
Authors	**Yash Bhalgat**, Iro Laina, João F. Henriques, Andrew Zisserman, Andrea Vedaldi
Publication status	Published
Publication details	[Spotlight] Neural Information Processing Systems (NeurIPS), 2023.

Student Confirmation

Student name	Yash Bhalgat	
Contribution to the paper	First-author contribution: • conception of research ideas • design and implementation of models • writing and presentation of the paper	
Signature and Date		Nov 4th 2025

Supervisor Confirmation

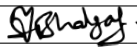
By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Dr. João F. Henriques	
Supervisor comments		
Signature and Date		5/11/2025

Statement of Authorship for the paper "N2F2: Hierarchical Scene Understanding with Nested Neural Feature Fields" in Chapter 4.

Paper title	N2F2: Hierarchical Scene Understanding with Nested Neural Feature Fields
Authors	**Yash Bhalgat**, Iro Laina, João F. Henriques, Andrew Zisserman, Andrea Vedaldi
Publication status	Published
Publication details	European Conference on Computer Vision (ECCV), 2024.

Student Confirmation

Student name	Yash Bhalgat	
Contribution to the paper	First-author contribution: • conception of research ideas • design and implementation of models • writing and presentation of the paper	
Signature and Date		Nov 4th 2025

Supervisor Confirmation

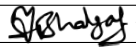
By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Dr. João F. Henriques	
Supervisor comments		
Signature and Date		5/11/2025

Statement of Authorship for the paper "3D-Aware Instance Segmentation and Tracking in Egocentric Videos" in Chapter 5.

Paper title	3D-Aware Instance Segmentation and Tracking in Egocentric Videos
Authors	**Yash Bhalgat***, Vadim Tschernezki*, Iro Laina, João F. Henriques, Andrea Vedaldi, Andrew Zisserman
Publication status	Published
Publication details	Asian Conference on Computer Vision (ACCV), 2024.

Student Confirmation

Student name	Yash Bhalgat	
Contribution to the paper	Co-first author contribution: • joint conception of research ideas • design and implementation of tracking models, experiments and evaluation protocols • writing and presentation of the paper	
Signature and Date		Nov 4th 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	Dr. João F. Henriques	
Supervisor comments		
Signature and Date		5/11/2025