Unsupervised Scene Understanding



Supervised by Professor Andrea Vedaldi
and Professor Christian Rupprecht

Aleksandar Shedritski

Magdalen College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Hilary 2025

Acknowledgement

It's not about the journey or the destination; it's about the company along the way.

Foremost, I want to thank my amazing supervisors, Andrea Vedaldi and Christian Rupprecht, for their wisdom, constructive criticism, and for helping me navigate the crazy field of AI research. I feel super lucky to have been supervised by the two of you!

I was very fortunate to do my DPhil around some very outstanding people. First, I am very grateful to the people I got to collaborate with on various side-quest research projects: Tim Franzmeyer, Paul Engstler, Max Bain, Iro Laina, Hannah Rose Kirk, Jana Zeller, Siobhan Hall, Odhran O'Donoghue. I would also like to thank everyone in VGG (Very Good Group) for being such a friendly and smart bunch, especially Laurynas, Tim, Luke, Stan, Paul, Ragav, Max, Sagar, Prannay, Rhydian, Tengda, Chuhan, Dominik, Charig, Vadim – really looking forward to seeing how much you all achieve. I would also like to thank Ash for always promptly resolving my infra problems.

Next, I would like to thank everyone from the AIMS CDT, especially those in the 2020 cohort for the great support group they have been, and those in the previous cohorts for providing very useful, and sometimes cynical, advice. I am especially grateful to Wendy Poole, who elevated my PhD experience by always making sure we could enjoy social events, and helping with all things admin and travel.

I would also like to thank all other people I got to meet during my time in Oxford. I was fortunate to have some great housemates, especially Martin, Jacob, Will, Sally, Irena, Mahdi and Maria who will always stay close to my heart. A large chunk of my "free" time I spent with the Oxford AI Society – it was great working with some amazing OxAI presidents along the years—Odhran, Elias, Roxana, Jessie, Ivan. I am especially grateful to Odhran for introducing me to OxAI. I am also very thankful to the Magdalen MCR for being such a vibrant and welcoming community, especially during Covid, and for always providing a bit of drama when life gets too boring (e.g. when we controversially made national news). I am particularly grateful to Gabi and Martin for sharing the bar

and food responsibilities with me, which was a great time.

I spent an amazing summer interning in Google DeepMind in London, and would like to thank my mentor Olivia Wiles, as well as other people I got to collaborate with: Aida, Nelly, Ira, Sahand, Isabella, Emmanuele, Ivana. I was also lucky to meet other interns that made my time at DeepMind a lot of fun, especially Federico, Roberto, Jaiden, Antoine, Harsh, and Noemi.

I would like to thank my high school friends, who I was very lucky to have kept in touch with, especially those who chose to do a PhD and would always be happy to rant with me: Yana, Marieta, Ribnishki and Tzvetan. And also to those who didn't—Kuny, Slavi, Ivan, Anton, Sofi.

Lastly, to those closest to me. I am very grateful to my parents, Poli and Sasho, for raising me up, and for always putting up with me, no matter how moody or grumpy I may be. To my sister, Elena, for looking up to me, which always motivates me to be my best. And to my lovely girlfriend, Yana, who manages to make me more humble, but also unwaveringly believes in me. Thank you!

Abstract

Foundation vision models, trained on vast amounts of data, underpin the majority of work on image understanding and generation. Image-only models like DINO and image-text models like CLIP learn powerful representations that are adapted and used in state-of-the art segmentation, captioning, image and 3D generation, and more. Since these models have "seen" a large part of the images on the internet, they possess large latent knowledge and exhibit various emergent abilities. This thesis explores methods to use these models to solve various computer vision tasks they have not been trained on, in an unsupervised manner, without using any manually-labelled data.

First, we explore the problem of correspondences, which is fundamental to scene understanding. We begin by developing an unsupervised method for establishing image-to-image matches, where we adapt foundation model features for this task using geometric priors. Next, we study image-to-3D correspondences, and find that by reducing the problem to that of image-to-image matching, we can repurpose off-the-shelf unsupervised methods. Doing so, we develop a method that outperforms state-of-the-art supervised methods, without using any human annotations.

Second, we discover an emergent ability of CLIP models, where by overlaying a visual marker, such as a red circle, over an image, we focus CLIP's vision encoder to that region. We term that process "visual prompt engineering", systematically analyse that emergent behaviour, and find it opens up a wide range of applications. Next, we propose a method to automatically discover such visual prompts, and observe that they consistently look like red contours, strikingly resembling human-engineered ones.

Finally, we apply similar ideas to 3D scene generation. Using prompt engineering techniques, we repurpose image generators and 3D object generators to generate

3D worlds with no training or optimization. The 3D worlds we generate are fully coherent, can be explored, and can be arbitrarily extended in size.

Keywords – Unsupervised learning, Foundation models, VLMs, prompt engineering

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.

Aleksandar Shtedritski, June 2025.

Contents

1 Introduction and Background **8**

1.1 Motivation 8

1.2 Key Ideas and Literature Review 10

1.2.1 Emerging Properties of Deep Features. 10

1.2.2 Emergent Behaviours of Foundation Models 11

1.3 Thesis Outline and Contributions 13

1.3.1 Publications 15

2 Learning Universal Semantic Correspondences With No Supervision and Automatic Data Curation **17**

2.1 Introduction 19

2.2 Related work 22

2.3 Method 24

2.3.1 Scoring matches 25

2.3.2 Learning a matcher by adapting features 26

2.3.3 Incorporating geometric constraints 28

2.3.4 Choosing images and points to match 28

2.4 Experiments 29

2.4.1 Datasets and Metrics 29

2.4.2 Implementation Details 30

2.4.3 Results . . . 32
2.5 Conclusions . . . 36
2.6 Appendix . . . 37
2.6.1 Discussion of the Architecture . . . 37
2.6.2 Implementation details . . . 39
2.6.3 Qualitative Evaluations . . . 41

3 SHIC: Shape-Image Correspondences with No Keypoint Supervision 42

3.1 Introduction . . . 44
3.2 Related work . . . 46
3.3 Method . . . 47
3.3.1 Canonical surface maps . . . 48
3.3.2 Unsupervised image-to-image correspondences . . . 49
3.3.3 Unsupervised image-to-template correspondences . . . 49
3.3.4 Unsupervised canonical surface maps . . . 51
3.3.5 Increasing the realism of synthetic data . . . 54
3.3.6 Learning formulation . . . 54
3.4 Experiments . . . 55
3.4.1 Implementation details . . . 55
3.4.2 Training data . . . 56
3.4.3 Evaluation . . . 57
3.5 Conclusion . . . 60
3.6 Appendix . . . 61
3.6.1 Limitations . . . 61
3.6.2 Additional implementation details . . . 61
3.6.3 Additional ablations . . . 63
3.6.4 Qualitative examples . . . 64

3.6.5 Failure case . . . 64

4 What Does CLIP Know About a Red Circle? Visual Prompt Engineering for VLMs 69

4.1 Introduction . . . 70
4.2 Related work . . . 73
4.3 Method . . . 76
4.3.1 Prompt engineering . . . 76
4.3.2 Visual prompting via marking . . . 77
4.3.3 Tasks . . . 78
4.4 Experiments . . . 80
4.4.1 Naming Keypoints . . . 80
4.4.2 Localizing Keypoints . . . 83
4.4.3 Referring Expression Comprehension . . . 85
4.4.4 Model biases and ethics . . . 86
4.5 Conclusions . . . 88
4.6 Appendix . . . 89
4.6.1 Datasets . . . 89
4.6.2 Additional implementation details . . . 91
4.6.3 Qualitative evaluations . . . 92

5 Highlight: Learning Visual Prompts for Vision-Language Models 100

5.1 Introduction . . . 102
5.2 Related Work . . . 103
5.3 Method . . . 105
5.3.1 Visual prompting . . . 105
5.3.2 Overlaying a visual prompt. . . . 106
5.3.3 Highlight: Learning a Visual Prompt . . . 107
5.4 Experiments . . . 109

5.4.1 Datasets . . . 109
5.4.2 Implementation details . . . 111
5.4.3 Results . . . 111
5.5 Conclusion . . . 116
5.6 Appendix . . . 117
5.6.1 Adding Heuristical Spatial Relations . . . 117
5.6.2 Transfer between models . . . 118
5.6.3 Comparison to FGVP . . . 118
5.6.4 Implementation details . . . 119
5.6.5 Additional qualitative results . . . 120

6 SynCity: Training-Free Generation of 3D Worlds 122
6.1 Introduction . . . 124
6.2 Related Work . . . 127
6.3 Method . . . 129
6.3.1 Prompting the Language Model . . . 130
6.3.2 Prompting the 2D Generator . . . 130
6.3.3 Prompting the 3D Generator . . . 132
6.3.4 3D Blending . . . 134
6.4 Experiments . . . 136
6.4.1 Ablations . . . 137
6.4.2 Qualitative Results . . . 139
6.5 Conclusion . . . 140
6.6 Appendix . . . 140
6.6.1 Language model prompting details . . . 140
6.6.2 2D prompting details . . . 141
6.6.3 3D geometry validation details . . . 141
6.6.4 3DGS post-processing details . . . 142

6.6.5 Ablation details . . . 143
6.6.6 Additional qualitative results. . . . 144
6.6.7 Limitations . . . 145

7 Discussion **150**
7.1 Summary and Impact . . . 150
7.1.1 Unsupervised Correspondences . . . 150
7.1.2 Visual prompting . . . 151
7.1.3 3D Scene Generation . . . 152
7.2 Future Work . . . 153
7.3 Conclusion . . . 153

References **154**

Chapter 1

Introduction and Background

In this chapter, we outline and motivate the research questions of this thesis, and give an overview of the existing literature.

1.1 Motivation

The current era in computer vision, and deep learning in general, is defined by the use of vast amounts of data used to train large generalist models. Datasets of ever-increasing scale are used, for instance CLIP [Radford et al. 2021] was trained on 400M, Stable Diffusion [Rombach et al. 2021] on 2B, and SigLIP-2 [Tschannen et al. 2025] on 10B text-image pairs. The vast quantity of images in these datasets means they inevitably capture the large diversity of the web. However, the sheer size of these datasets also means that it is extremely difficult to fully understand their content. For example, LAION-5B [Schuhmann et al. 2022] contained multiple extremely harmful images, which took the community more than a year to discover [Bloomberg News 2023], after multiple leading models were trained on it [Rombach et al. 2021; Cherti et al. 2022; Q. Sun et al. 2024]. Having said that, the diversity of these datasets also capture modes of data that are actually desirable, although we do not expect them a priori. For instance, CLIP [Radford et al. 2021] exhibits multilingual capabilities, can perform optical character recognition, and is robust to distributions shifts, such as sketches or cartoons. It is likely that none of these abilities was specifically targeted, but emerged from its training data.

The latent "knowledge" of large models pretrained on such datasets makes them

very suitable to build on when developing scene understanding systems. By fine-tuning them or using them as backbones, models such as CLIP [Radford et al. 2021] and DINO [Caron et al. 2021a] find applications in various downstream tasks, such as open-vocabulary segmentation [S. Liu et al. 2023] or image captioning [Junnan Li et al. 2022]. In fact, the majority of state-of-the-art methods in computer vision tasks depend on such foundation models in some way. However, training these methods still requires large amounts of task-specific human-labelled data, which is expensive to acquire and is not scalable. Given that, developing systems that can interpret visual scenes with *no* human supervision, that utilise the powerful representations and latent knowledge of internet-scale models, is an area of particular interest.

Foundation models have been invaluable for progress in unsupervised tasks, from semantic keypoint matching [Amir et al. 2021b; Junyi Zhang et al. 2023a] to semantic segmentation [Y. Wang et al. 2022b; Melas-Kyriazi et al. 2022a; Hamilton et al. 2022; Y. Wang et al. 2022b]. However, unsupervised approaches are still far behind in performance compared to supervised ones. Furthermore, unsupervised methods traditionally are difficult to tune and rely on a large number of heuristics, making them dataset-specific and not generalizable to different domains. On the other hand, Large Language Models (LLMs) easily exhibit emergent properties through prompting techniques [Radford et al. 2019; T. Brown et al. 2020] and can zero-shot or few-shot adapt to new tasks.

In light of these challenges, this thesis argues that the powerful representations learnt by large pretrained models can contribute to better unsupervised scene understanding systems. Furthermore, by tapping into the latent knowledge resulting from the diverse data these models have been trained on, we can develop methods to "prompt" them, eliciting useful behaviours. The next chapters of this thesis will look into our approach in detail, and discuss the ways in which it can shape the future of using foundation vision models.

1.2 Key Ideas and Literature Review

1.2.1 Emerging Properties of Deep Features.

Self-Supervised Features. Deep features have been shown to correlate well with human perception [Gatys et al. 2016; Johnson et al. 2016; Richard Zhang et al. 2018] and to encode various types of visual information, from low-level concepts such as colors and edges to high-level concepts such as semantics [Olah et al. 2017]. More recently, researchers have observed a number of emerging properties of the deep features of large-scale pretrained models. In particular, the features of self-supervised vision transformers (ViTs) [Dosovitskiy et al. 2020] were found to have more interesting properties than the self-supervised CNNs, or their supervised equivalents. For example, DINO [Caron et al. 2021a] features were shown to contain explicit semantic information, which naturally leads to unsupervised semantic segmentation methods [Melas-Kyriazi et al. 2022a; Y. Wang et al. 2022b], and MAE [He et al. 2021] can separate instances, and are thus good for unsupervised instance segmentation [Engstler et al. 2023]. Furthermore, self-supervised ViT features were found to be good local descriptors that can be used for zero-shot part segmentation and point correspondences [Amir et al. 2021b]. More recently, deep features from text-to-image diffusion models such as Stable Diffusion [Rombach et al. 2021] have also been shown to perform well on various tasks such as segmentation and correspondences [Junyi Zhang et al. 2023a; Hedlin et al. 2023; L. Tang et al. 2023; J. Tian et al. 2024].

Unsupervised Visual Reasoning. The self-supervised features such as those of DINO are not explicitly trained for solving tasks such as semantic segmentation or correspondences. This has motivated research into methods for *adapting* these features in an unsupervised way, in order to make them more suitable for specific tasks and close the gap with supervised approaches. For example, STEGO [Hamilton et al. 2022] and CutLER [Xudong Wang et al. 2023] distill DINO features in order to learn semantic segmentation. In Chapter 2, we present [Shtedritski et al. 2023a], a method for distilling DINO features for semantic correspondences. We do this by predicting adapted "prototype" features that better capture the semantics and locations of the keypoints in the image.

Generalization Properties. One of the motivations behind using these large-scale pretrained features for different computer vision tasks is that foundation models have been trained on a very diverse set of data, which would suggest some emergent generalization properties. For instance, CLIP has been shown to be capable of zero-shot classifying images from different domains, such as drawings or sketches [Vinker et al. 2022; Radford et al. 2021], pointclouds [Renrui Zhang et al. 2022] or 3D shapes [Dan Song et al. 2025]. In Chapter 3, we present [Shtedritski et al. 2024], where the generalization properties of the deep features of DINO and Stable Diffusion enable finding dense correspondences between natural images and renderings of 3D meshes. The concurrent work of [Caraffa et al. 2024] apply similar ideas to the task of 6D pose estimation.

Shortcomings. Despite the power of these features to solve many computer vision tasks training-free, or with little unsupervised adaptation, they remain very ad-hoc, and methods and findings do not generalize well to new model classes. For example, when using ViT features, one can select features across the different transformer layers and facets (key, query, value, token). When using DINO-v1 [Caron et al. 2021a] for semantic correspondences, the best features are the keys from layer 9 [Amir et al. 2021b]. However, for DINO-v2 [Oquab et al. 2023], the best features are the tokens from layer 11, and the best-performing features from DINO-v1, the keys from layer 9 perform more than 2× worse [Junyi Zhang et al. 2023a]. Similarly, when extracting features from diffusion models, in addition to the layer and facet, one has to also select a denoising timestep, and several studies show that the optimal timestep varies for different layers, facets, and more importantly, tasks [Zhan et al. 2024; Junyi Zhang et al. 2023a]. This means that most methods manually tune these hyperparameters [Junyi Zhang et al. 2023a; L. Tang et al. 2023] or learn them [G. Luo et al. 2023b], which undermines the unsupervised nature of these methods. Next, we explore prompting as a more general way of interacting with pretrained models, by treating them as black boxes.

1.2.2 Emergent Behaviours of Foundation Models

Emergent Behaviours in LLMs. Emergent *behaviours* were observed in LLMs, such as GPT-2 and GPT-3 [Radford et al. 2019; T. Brown et al. 2020], where the model could solve tasks it was not explicitly trained on, e.g. translate a phrase from

English to French in a zero-shot manner. Since the training data for these models was not specifically curated for that, it has been suggested that these behaviours are learnt from the vast amount of web-scraped training data. For instance, [Radford et al. 2019] find naturally occurring demonstrations of translations between languages in the pre-training corpus of GPT-2. Similarly, by prompting an LLM to "think step by step" [Kojima et al. 2022], or by encouraging it to output a reasoning trace [J. Wei et al. 2022], one can elicit the LLM to reason, despite not explicitly being trained for that.

Emergent Behaviours in Vision Models. Similarly to language models, vision (or vision-language) models exhibit emergent behaviours when prompted accordingly. For example, [Radford et al. 2021] observed that CLIP can perform zero-shot classification, when the textual descriptions of all possible classes are engineered into text prompts, and CLIP "selects" the most compatible text prompt to a given image. Similarly to LLMs, models like Flamingo [Alayrac et al. 2022] can be few-shot prompted to solve new tasks, where a few demonstrations of the task being solved are shown. In [Bar et al. 2022], such few-shot behaviours are explicitly learnt from data, where an inpainting model is trained on results figures from computer vision papers, and can at test time "copy" a pixel-to-pixel task, like semantic segmentation or pose estimation. Such models can also one-shot adapt to new tasks, unseen during training [Xinlong Wang et al. 2023]. In Chapter 4, we present [Shtedritski et al. 2023b], discovering an emergent ability of CLIP, where it can reason about visual markers, such as red circles, overlaid on top of images, and in Chapter 5 we present [Zeller et al. 2025], a method to automatically discover such visual markers. Similar abilities have since been observed in multimodal LLMs, such as GPT-4o [Achiam et al. 2023], which can be prompted with arrows, circles, or other annotations [Bubeck et al. 2023], although such commercial models have likely been fine-tuned to reason about visual markers. In addition to red circles, visual markers such as outlines [Lingfeng Yang et al. 2024] and circles with a learnable color [Rezaei et al. 2024] also elicit these emergent behaviours.

Beyond Images. The use of foundation models has been especially influential in the field of 3D generation, where due to the lack of large-scale 3D data, 2D generative models such as Stable Diffusion [Rombach et al. 2021] have been

used as 2D priors. DreamFusion [Poole et al. 2023] was among the first to use a state-of-the-art 2D image generator to enable text-to-3D object synthesis, and RealFusion [Melas-Kyriazi et al. 2023] applied similar ideas to image-to-3D generation. Since, large-scale datasets such as Objaverse [Deitke et al. 2023a] and Objaverse-XL [Deitke et al. 2023b] have enabled methods like Zero-1-to-3 [R. Liu et al. 2023b] and Trellis [Xiang et al. 2024b] that can generate high quality 3D objects from images or text. However, to generate entire 3D worlds that problem remains: there is no diverse and large dataset of 3D scenes, and such a dataset would be especially difficult to curate. In Chapter 6, we present SynCity [Engstler et al. 2025a], a method for generating entire 3D worlds with no training or optimisation, that leverages foundation 2D generators (Flux [Labs 2023]) and 3D object generators (Trellis [Xiang et al. 2024b]), which are repurposed using novel prompt engineering techniques.

1.3 Thesis Outline and Contributions

In this section, we outline the contributions and structure of the thesis. We provide an overview of each chapter, and divide the thesis into three themes: (i) Unsupervised Correspondences, (ii) Visual Prompting, and (iii) 3D Scene Generation.

Unsupervised Correspondences

In this research area, we look at the problem of establishing correspondences between images or objects. This is a fundamental computer vision problem and a necessary step in numerous image analysis tasks, from image understanding to pose estimation and neural rendering. In particular, we focus on discovering *semantic* correspondences between pairs of objects.

In Chapter 2, we propose a framework for representation learning for semantic correspondences. Our key idea is to combine powerful pretrained features (e.g. DINO [Caron et al. 2021a]), which have good generalization, but poor spatial locality, with traditional geometric priors, which have good spatial locality, but poor generalization. We use these to derive a pair of complementary learning signals, which we use to train on *non-curated* dataset of images, such as ImageNet [Russakovsky et al. 2015]. Ours is the first method for point correspondences that learns from a general-purpose dataset without any manual curation of matching

image pairs, and train a single, universal, model, that we apply to all benchmarks datasets.

In Chapter 3, we consider the problem of establishing correspondences between an image and a 3D template, e.g. a mesh. We begin by reducing the image-to-3D matching problem to that of image-to-image matching. To do that, first we note that off-the-shelf unsupervised semantic correspondences methods can already match natural images to non-photorealistic renderings of a 3D object, which emulates the process of collecting manual annotations for this task. Next, we lift these image-to-image correspondences to 3D, and use them to train a high-quality canonical map for any object of interest. Our model is trained only on unsupervised pseudo-labels, yet it outperforms supervised methods on this task.

Visual Prompting

Under this research theme, we look at visual prompting as a way of interacting with Vision-Language Models (VLMs), revealing some interesting emergent abilities of foundation models.

In Chapter 4 we discover an emergent ability of CLIP, where simply by drawing a red circle (in practice, usually an ellipse) over a region in an image, we can direct the model's attention to that region. We show the power of this simple approach by achieving state-of-the-art in several unsupervised localization tasks. We thoroughly explore this ability of CLIP models, and find it only emerges for the largest models, and is most prominent when the color and shape of the marker are red and circle, respectively. Finally, we find some ethical concerns with such techniques, where by drawing red circles around people, they get zero-shot classified as missing or murderers, potentially due to biases in the training data.

Next, in Chapter 5 we question the intuitive and manually-engineered red circle. In particular, we design a learning framework, where we overlay a *learnable* visual marker over a region in the image, and train it such that the marker draws the attention of the model. We propose several, supervised and unsupervised, flavours of our approach, and find that the learnable visual prompts outperform the manually engineered ones. Interestingly, however, we find that the automatically learnt visual prompts consistently look like red contours, strikingly resembling the manually engineered ones, further suggesting that this emergent ability is learnt from

such examples in the training data of the models.

3D Scene Generation

Finally, we use prompt engineering in order to repurpose an off-the-shelf image inpainter and an object-centric image-to-3D model to generate coherent 3D worlds. In particular, we find that if we take a local part of the 3D world that is placed on top of a square slab, or a tile, it can be reconstructed by the image-to-3D model, as it is being interpreted as an object. With that in mind, we prompt the image inpainter to sequentially generate parts of the 3D world, which are the reconstructed in 3D and combined to form the 3D world in a manner that preserve consistency by blending neighbouring tiles. Combining the artistic versatility of 2D models with the geometric precision of 3D models, we generate entire 3D scenes that are diverse, coherent, and rich in detail.

1.3.1 Publications

Chapters 2 to 6 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 Learning Universal Semantic Correspondences with No Supervision and Automatic Data Curation

Aleksandar Shtedritski, Andrea Vedaldi, Christian Rupprecht

In *International Conference on Computer Vision Workshops*, 2023

Chapter 3 SHIC: Shape-Image Correspondences with No Keypoint Supervision

Aleksandar Shtedritski, Christian Rupprecht, Andrea Vedaldi

In *European Conference on Computer Vision*, 2024

Chapter 4 What Does CLIP Know About a Red Circle? Visual Prompt Engineering for VLMs

Aleksandar Shtedritski, Christian Rupprecht, Andrea Vedaldi

In *International Conference on Computer Vision*, 2023

Chapter 5 Highlight: Learning Visual Prompts for Vision-Language Models

Jana Zeller, **Aleksandar Shtedritski**, Christian Rupprecht

In *Computer Vision and Pattern Recognition Conference Workshops*, 2025

Chapter 6 SynCity: Training-Free Generation of 3D Worlds

Paul Engslter*, **Aleksandar Shtedritski***, Iro Laina, Christian Rupprecht, Andrea Vedaldi (*Equal contribution)

In *International Conference on Computer Vision*, 2025

Publications not included In addition, I have written those during my DPhil, but have excluded them due to their lower relevance to the main topic of this thesis.

- **Bias out-of-the-box: An empirical analysis of intersectional occupational biases in popular generative language models**

  Hannah Rose Kirk, Yennie Jun, Filippo Volpin, Haider Iqbal, Elias Benussi, Frederic Dreyer, **Aleksandar Shtedritski**, Yuki Asano

  NeurIPS, 2021

- **BioPlanner: automatic evaluation of LLMs on protocol planning in biology**

  Odhran O'Donoghue, **Aleksandar Shtedritski**, John Ginger, Ralph Abboud, Ali Essa Ghareeb, Justin Booth, Samuel G Rodriques

  EMNLP, 2023

- **Visogender: A dataset for benchmarking gender bias in image-text pronoun resolution**

  Siobhan Mackenzie Hall, Fernanda Gonçalves Abrantes, Hanwen Zhu, Grace Sodunke, **Aleksandar Shtedritski**, Hannah Rose Kirk

  NeurIPS Datasets and Benchmarks, 2023

- **HelloFresh: LLM Evaluations on Streams of Real-World Human Editorial Actions across X Community Notes and Wikipedia edits**

  Tim Franzmeyer*, **Aleksandar Shtedritski***, Samuel Albanie, Philip Torr, João F Henriques, Jakob N Foerster (*Equal contribution)

  ACL, 2024

Chapter 2

Learning Universal Semantic Correspondences With No Supervision and Automatic Data Curation

The paper was presented at the ICCV 2023 LIMIT Workshop.

Learning Universal Semantic Correspondences with No Supervision and Automatic Data Curation

Aleksandar Shtedritski Andrea Vedaldi Christian Rupprecth

Visual Geometry Group, University of Oxford

{suny, vedaldi, rupprecht}@robots.ox.ac.uk

September 29, 2025

Abstract

We study the problem of learning semantic image correspondences without manual supervision. Previous works that tackled this problem rely on manually curated image pairs and learn benchmark-specific correspondences. Instead, we present a new method that learns universal correspondences once, from a large image dataset, and without using any manual curation. Despite their generality and despite using less supervision, our universal correspondences still outperform prior works, unsupervised and weakly supervised, in most benchmarks. Our approach starts from local features extracted by an unsupervised vision transformer, which obtain good semantic but poor geometric matching accuracy. It then learns a Transformer Adapter which improves the geometric accuracy of the features, as well as their compatibility between pairs of different images. The method combines semantic similarity with geometric stability obtained via cycle consistency and supervision via synthetic transformations. We use these features to also select pairs of matching images for training the unsupervised correspondences.



Figure 2.1: **Unsupervised Universal Semantic Correspondences.** Correspondences found by our method on images across different classes. The method is able to find meaningful correspondences across object categories and poses without any supervision, and with no manual curation of the training data. Best viewed digitally and in color.

2.1 Introduction

Establishing image correspondences is a necessary step in numerous image analysis tasks, from image understanding to pose estimation and neural rendering. In this work, we tackle the problem of discovering *semantic* correspondences between pairs of images containing different objects or object types (Fig. 2.1), which is crucial for high-level visual reasoning in diverse environments. Due to the cost of collecting manual annotations for this task, and the absence of large-scale datasets for it, we aim at learning a correspondence predictor without manual supervision.

Most unsupervised approaches for learning correspondences are based on enforcing a form of cycle consistency between images [Truong et al. 2022; Truong et al. 2021; I. Rocco et al. n.d.; I. Rocco et al. 2017]. However, cycle consistency can only tell when correspondences are poor, but cannot suggest or induce meaningful correspondences by itself [Truong et al. 2021]. Hence, cycle consistency needs to be combined with additional unsupervised learning signals, such as synthetic transformations of the images, from which correspondences can be induced.

Despite progress, it remains difficult to discover correspondences that generalize well to new categories and datasets. A clear sign of this limitation is that prior methods rely on training separate models for each downstream task or benchmark, which often requires domain-specific data curation [Xin Li et al. 2021; Truong et al. 2021; Truong et al. 2022; I. Rocco et al. n.d.; I. Rocco et al. 2017; Peebles et al. 2022]. In this paper, we develop an unsupervised method that (1) **learns from a single, non-curated dataset** like ImageNet (2) **universal correspondences** that generalize to multiple benchmark datasets. Because of its generality, our correspondence predictor is more versatile than previous ones; furthermore, because training does not require data curation, it can be trained much more easily and cheaply on large datasets.

The first limitation that we address is the one of generalization. Approaches based on synthetic transformations of single images generalize poorly, as they can only learn 'semantically local' correspondences. For example, both a bird and a tiger have a head and a neck, but it is difficult to discover cross-category correspondences by applying random synthetic transformations to either (Fig. 2.1).

In order to address this limitation, we consider the 'dual' nature of correspondences and concepts. A coherent system of correspondences partitions 2D image points in equivalence classes, each corresponding to a certain concept or abstraction. For example, in multi-view geometry we often put all 2D projections of the same 3D physical point in correspondence, so the physical point can be thought of as the grouping of all its image projections. Likewise, the concept of 'neck' can be thought of as the grouping of all 2D points that correspond to the neck of all animals in all images.

Because of this duality, discovering semantic correspondences is closely related to discovering semantic parts. The latter has been boosted significantly by recent discoveries in self-supervised learning. Models such as DINO [Caron et al. 2021a], that combine transformers (ViT) with deep invariant clustering, have been found to extract features that are well correlated with semantic object parts, all without using any manual supervision. Follow-up works [Tumanyan et al. 2022a; Amir et al. 2021a] have readily noted that such features can also establish semantic correspondences between images, and these *generalize* particularly well. Even so, methods such as ViT-DINO are *not explicitly designed* for this task. For example,

we show empirically that the induced correspondences tend to confuse multiple occurrences of the same semantic part, e.g., by grouping all legs of all animals in a single cluster (Fig. 2.2). Hence, our goal is to improve the geometric accuracy and robustness of these correspondences. To this end, we make three technical contributions.

Our first contribution is to combine the information that is obtained from ViT-DINO features (poor spatial locality but good generalization) and traditional cycle-consistency (good spatial locality but limited generalization). We derive from these a pair of complementary learning signals and distill a high-quality point matcher from them. Experiments clearly show the benefit of this combination, which significantly reduces the spatial uncertainty of the matches.

Our second contribution is a powerful formulation to predict correspondences. We cast this as retrieving a point in the target image given a point in the source, based on comparing descriptors derived from the ViT-DINO features. We learn a transformer network that takes the pre-trained ViT feature of a point in the source image and adapts it to generate a query descriptor to be matched in the second image. We also pass to the transformer a global descriptor of the target image. In this manner, the transformer can improve the locality of the source descriptor and adapt it to *bridge appearance differences* (e.g., matching a white dog to a black one), further boosting quality.

Our third contribution is an approach to learning unsupervised correspondences using a non-curated dataset such as ImageNet, which addresses the limitation that prior works require manually-curated image pairs. The challenge here is that not all pairs of source and target images can be matched. For example, it makes little sense to match points between a broom and an elephant. Prior work often uses curated pairs of images for training, ensuring their semantic compatibility. Here, we show that it is possible to fully *automatize pair selection* by finding images with sufficiently close global ViT descriptors. Our is the first method to learn point correspondences across different images from a general-purpose dataset without any curation of the matching image pairs.

We test our method against prior work on unsupervised image matching and show large gains in accuracy compared to the baseline ViT-DINO features in numerous

benchmark datasets without the need to train separate models. Furthermore, for the most difficult datasets, we also obtain state-of-the-art performance when compared to all prior works on unsupervised keypoint matching, even when compared to weakly supervised methods.

2.2 Related work

Supervised Semantic Correspondences. Most approaches to semantic correspondences are based on cost volumes over image features, which are subsequently used to estimate displacements represented as geometric transformations [J. Chen et al. 2019; Seungryong Kim et al. 2018; Ignacio Rocco et al. 2017; Ignacio Rocco et al. 2018] or flow fields [Seungryong Kim et al. 2019; Truong et al. 2020b; Truong et al. 2021] from source to the target image. The aggregation of the cost volumes has received particular attention, where 4D convolutions are traditionally used [Ignacio Rocco et al. 2020; Shuda Li et al. 2020; J. Y. Lee et al. 2021]. More recently, transformers have been shown to be very effective for cost aggregation as well [S. Hong et al. 2022; W. Jiang et al. 2021; S. Cho et al. 2021; S. Cho et al. 2022]. Differently from these works, we aggregate features using *global* rather than *local* descriptors, which is necessary when there are large appearance or pose changes between the pair of images. Other approaches focus on combining multilevel features [Ufer and Ommer 2017; J. Min and M. Cho 2021; J. Min et al. 2020; D. Zhao et al. 2021] or using optimal transport [Yanbin Liu et al. 2020; Sarlin et al. 2020]. Using pseudo labels and consistency across geometric transformations, [J. Kim et al. 2022] show that unsupervised objectives can improve the performance of supervised semantic correspondence methods.

Weakly Supervised and Unsupervised Semantic Correspondences. Early unsupervised approaches artificially warp images and predict the transformation [J. Chen et al. 2019; Ignacio Rocco et al. 2017; Truong et al. 2020b; Melekhov et al. 2019; Truong et al. 2020a], but that usually results in poor generalization. Recently, GANgealing [Peebles et al. 2022] proposed to use GAN supervision for dense visual alignment. It achieves good performance, but it is limited to classes where a pre-trained GAN is available. Similarly, [Ofri-Amar et al. 2023] learn a 2D atlas for a given class, but their method only works for a limited number of classes

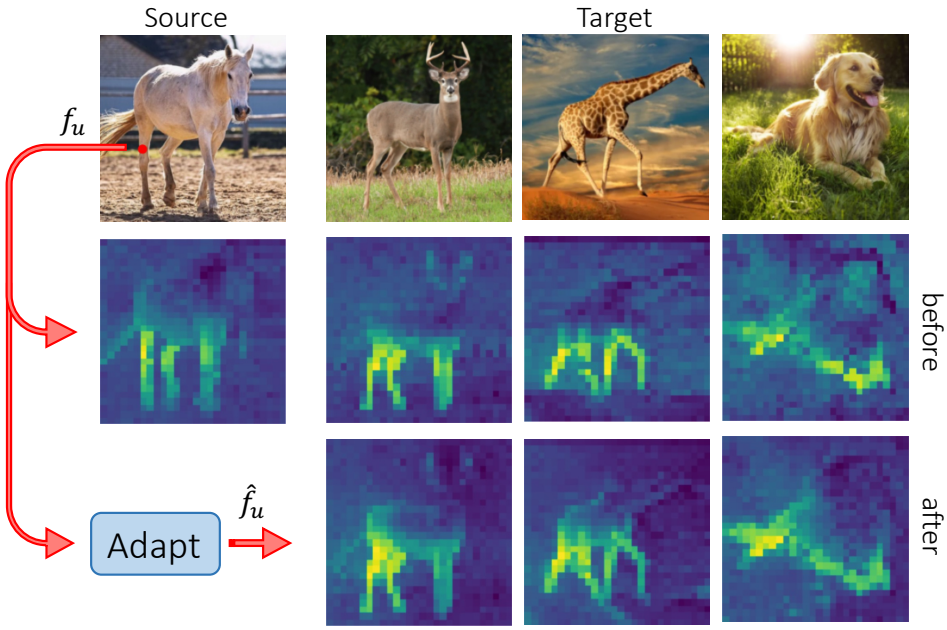


Figure 2.2: **Feature Similarity.** The heatmap is generated by computing the cosine similarity between a query feature in the leftmost image within the same image and with different ones, using ViT-DINO features. We see that the features capture the image semantics well (the heatmap highlights the legs), but do not provide precise localization (all legs are highlighted). We adapt the query feature f_u to an improved one, $\hat{f}_u$, that is more precise.

and needs careful manual curation of the images used. [T. Zhou et al. 2016] learn semantic correspondences using 3D cycle consistency, requiring 3D CAD models during training. In the weakly supervised case, where pairs of positive and negative matches are given, various auxiliary losses can be used [I. Rocco et al. 2018; Shuaiyi Huang et al. 2019; J. Min et al. 2020]. Another form of supervision is using forward-backward consistency [Jeon et al. 2018] and cycle consistency [Truong et al. 2021; Truong et al. 2022]. Truong *et al.* [Truong et al. 2021] show that some cycles, such as a simple forward-backward cycle, can lead to a degenerate solution where the model simply predicts the identity transformation, and suggest using a particular form of a triplet cycle instead. Some methods use pseudo matches [J. Kim et al. 2022] or discard inconsistent matches [I. Rocco et al. n.d.] using somewhat arbitrary thresholds. Instead, we propose a contrastive loss that deals with this in a principled manner. Finally, while all these methods use human-annotated pairs of images, our method is able to train with automatically discovered image pairs. A closely related line of work to unsupervised semantic correspondences is keypoint discovery, where the goal is to discover semantically meaningful keypoints without supervision [Jakab et al. 2018; Jakab et al. 2020; T. D. Kulkarni et al.

2019; Ryou and Perona 2021; Y. Zhang et al. 2018].

Self-Supervised ViT Features. Self-supervised ViT features [Caron et al. 2021a] have been shown to be surprisingly good visual descriptors [Amir et al. 2021a], performing very well in the task of semantic correspondences, and others such as semantic segmentation, co-segmentation, etc. Recently, fine-tuning ViT-DINO features in a task-specific unsupervised way has shown great potential. [Hamilton et al. 2022] proposes STEGO, distilling ViT-DINO features for semantic segmentation. While its unsupervised objective clusters semantically similar features, here we do the opposite and learn features that lead to unique correspondences. Aygun *et al.* [Aygun and Mac Aodha 2022] use an equivariant representation learning approach, inspired by [Thewlis et al. 2017b; Thewlis et al. 2019; Karmali et al. 2022]. Similar to our work, the method learns to project self-supervised ViT features to a semantic correspondence space. However, they rely on task-specific datasets with image pairs, whereas we train a single model on unpaired data and evaluate on all correspondence datasets.

Feature Distillation. Feature distillation aims to compress the knowledge of a large and potentially complex model, usually called a teacher, into a smaller one, a student, while preserving performance [Buciluǎ et al. 2006]. Distillation can even improve the performance of the student compared to the teacher [Park et al. 2019; Bagherinezhad et al. 2018; Yim et al. 2017]. Li *et al.* [Xin Li et al. 2021] propose a student-teacher method for semantic correspondences, distilling a probabilistic teacher trained on synthetic data. Distillation of self-supervised ViT features has recently been used for semantic segmentation [Hamilton et al. 2022] and neural fields [Kobayashi et al. 2022; Tschernezki et al. 2022].

2.3 Method

Given a source image $I \in \mathbb{R}^{3\times H\times W}$, a target image $J \in \mathbb{R}^{3\times H'\times W'}$, and a point $u \in [0,1]^2$ representing a 2D location in the source image I, we wish to find the corresponding point $v \in [0,1]^2$ in the target image J. We formulate this task as learning a predictor $\hat{v}_\theta(u; I, J)$ with parameters θ that maps any point u in the source image to the corresponding point v in the target.

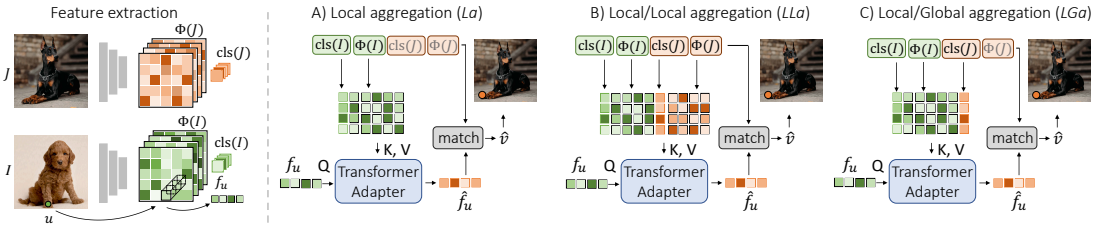


Figure 2.3: **Matching with a Transformer Adapter.** *Left:* Source and target images are encoded by a self-supervised ViT. We train a cross-transformer A that improves the query feature f_u for semantic matching. *Right:* Architecture variants.

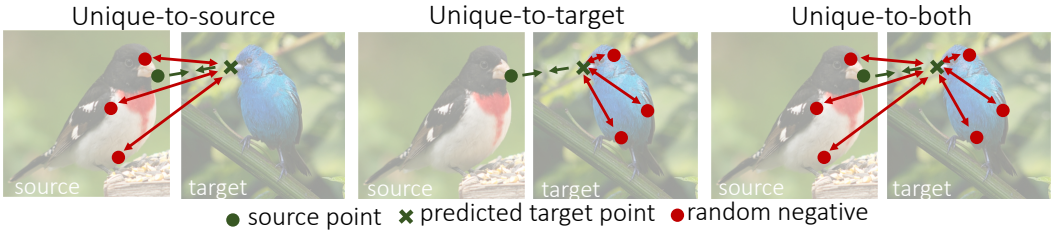


Figure 2.4: **Variants of the Contrastive Distillation Loss:** Depending on which images the negatives are sampled from, we define unique-to-source, unique-to-target, and unique-to-both losses. We find that the unique-to-both performs best.

To train the predictor $\hat{v}_\theta$ in an unsupervised manner, we score correspondences using ViT-DINO descriptors (Section 2.3.1) which we further improve via a transformer network (Section 2.3.2) that also incorporates geometric constraints (Section 2.3.3). We also automatically curate the training data (Section 2.3.4), a step which is usually done manually.

2.3.1 Scoring matches

Given the source image I, we use a self-supervised feature network to obtain a tensor $\Phi(I) \in \mathbb{R}^{D \times \frac{H}{K} \times \frac{W}{K}}$ of local feature descriptors (at a reduced image resolution). We use the symbol $f_u = \Phi_u(I) \in \mathbb{R}^D$ to denote the feature vector extracted from the source image I at location u via bilinear interpolation. We also use the symbol $g_v = \Phi_v(J) \in \mathbb{R}^D$ to denote the analogous feature vector for the target image J at location v.

We assess the 'compatibility' of points u and v in the two images by measuring the similarity of their feature vectors [Sarlin et al. 2020] using the function

$$s(f, g) = \exp\left(\frac{\langle f, g \rangle}{\tau \|f\| \|g\|}\right), \tag{2.1}$$

which takes larger values when the two feature vectors are more similar, and smaller values when they are less aligned.

We measure the quality of a match (u, v) via the *contrastive distillation loss* [Chopra et al. 2005]:

$$\mathcal{L}_d(v|u, I, J) = -\log \frac{s(f_u, g_v)}{\sum_{h \in \Phi(I) \cup \Phi(J)} s(h, g_v)}. \tag{2.2}$$

This loss states that the match (u, v) should receive a score larger than most matches (t, v), where t is any location in the source or target images — in Eq. (2.2) the symbol h denotes the feature vector of this arbitrary point t.

Discussion. As shown in Fig. 2.4, we choose Eq. (2.2) as the best among a few alternatives. These losses verify that the target location v matches back to other points t in a unique manner. This works in the reverse order of the predictor $\hat{v}_\theta$, which estimates v from the source u. This is on purpose: next, we construct $\hat{v}_\theta$ by approximately maximizing a similar score in the forward direction, so it would not make sense to also use it for verifying the match.

2.3.2 Learning a matcher by adapting features

Here we define the function $\hat{v}_\theta$ that maps the source location u to the corresponding target location $\hat{v}_\theta(u; I, J)$. The scoring function s of Eq. (2.1) provides a natural basis to establish such correspondences: we can define $\hat{v}$ to be the location in the target image J that maximizes the compatibility score $s(f_u, g_{\hat{v}})$ with the location u in the source image I. However, the features used to compute this score can be improved: Fig. 2.2 shows that matching ViT-DINO feature matches semantic *parts* well, but confuses keypoints. Hence, we learn a function that improves f_u, turning it in a good keypoint descriptor rather than a part descriptor.

We do so by replacing the descriptor f_u of the source point with an improved version $\hat{f}_u = A_\theta(f_u; I, J)$ computed by a learnable *adapter* function A_θ, detailed below. Then, we define the probability $p(v|u) \propto s(A_\theta(f_u; I, J), g_v)$ of matching the source point u to the target point v be proportional to the updated compatibility score. Finally, we define the match

$$\hat{v}_\theta(u; I, J) = \sum_v v \cdot p(v|u) \tag{2.3}$$

to be the expected value of the target v given the source u. Equation (2.3) reduces to maximising the compatibility between descriptors $\hat{f}_u$ and g_v when the score temperature τ tends to zero.

Adapter architecture. As discussed in Section 2.1, the goal of the adapter A_θ is to improve the descriptor f_u to disambiguate keypoints and to better match to the style of the target image J. To implement the adapter A, we use the fact that features are extracted by a ViT. This means that in addition to the spatial features $\Phi(I)$, the network extracts also a class token $\text{cls} \in \mathbb{R}^D$ which encodes the overall appearance of the image [Tumanyan et al. 2022b]. By using this token, the network A can adapt the appearance of the source features to the overall style of the target. In practice, A is a cross-attention transformer that takes in the spatial features $\Phi(I)$, the class tokens $\text{cls}(I)$ from the source image and $\text{cls}(J)$ from the target image. This results in a sequence of length $|\Phi(I)|+2$ as key and value inputs to the transformer. Finally, the query to the transformer is the source descriptor f_u (see Fig. 2.3).

Discussion. Most related work [Truong et al. 2022; Xin Li et al. 2021; Truong et al. 2020b; Ignacio Rocco et al. 2017; J. Lee et al. 2019] predicts an entire flow field at discrete pixel locations. In contrast, our design allows to evaluate a correspondence $\hat{v}_\theta(u; I, J)$ at individual source locations u with two advantages: (1) we only compute correspondences where they matter (typically the salient part of the image as seen in Section 2.3.4) and (2) u is not limited to exact pixel locations.

We also explored different designs for the transformer, shown in Fig. 2.3 and Section 2.4. Local aggregation (La) is the simplest, aggregating only source features. Local/Local aggregation (LLa) aggregates both source and target local features, as in e.g. COTR [W. Jiang et al. 2021]. Our Local/Global aggregation (LGa) combines local and global features from the source but only global features from the target. This is a bottleneck, forcing the adapter to predict what the query feature should look like in the target image.

Empirically, we found LGa (Fig. 2.3) to perform best. The La architecture, similar to [Aygun and Mac Aodha 2022], cannot account for the appearance of the target

features by design. The *LLA* architecture, can do so but learns a degenerate solution: the model can satisfy Eq. (2.2) by *choosing* a feature from $\Phi(J)$ that is similar to f_u but different enough from the other features in $\Phi(I)$. With *LGa*, however, the model needs to *predict* what the query feature should look like in the target image.

2.3.3 Incorporating geometric constraints

While ViT-DINO features generalize well, they lack spatial precision. The latter can be improved by training $\hat{v}_\theta$ to fit synthetic transformations and to be cycle-consistent. We follow the recommendations of [Truong et al. 2021] and implement a cycle involving three images: the source image I, the target image J, and a random synthetic transformation $G(I)$ (rotation, flipping, *etc.*) of the source image. Formally, given a point u in the source image, we define the loss:

$$\begin{aligned} \mathcal{L}_c(\hat{v}_\theta|u, I, J) &= a + \beta(b + c), \quad \text{where} \\ a &= \|G(u) - \hat{v}_\theta(\hat{v}_\theta(u; I, J); J, I)\|^2, \\ b &= \|G(u) - \hat{v}_\theta(u; I, G(I))\|^2, \\ c &= \|u - \hat{v}_\theta(G(u); G(I), I)\|^2. \end{aligned} \tag{2.4}$$

The first term a closes the cycle $I \to J \to I$ from source to the target image and back, whereas terms b and c fit the (known) synthetic correspondences in the direction I to $G(I)$ and $G(I)$ to I. We automatically balance the terms by setting $\beta = \text{sg}(a/(b + c))$ where sg is the stop-gradient operator.

2.3.4 Choosing images and points to match

Given a dataset of images $\mathcal{D}$ for training, we must choose a subset $\mathcal{P} \subset \mathcal{D} \times \mathcal{D}$ of image pairs $(I, J) \in \mathcal{P}$ that are meaningful to match. Randomly selecting pairs is unlikely to succeed as there are no meaningful correspondences between the vast majority of image pairs (e.g. car and fork, hamster and envelope, dog and plant, etc.). Prior work addressed this problem by only using the same image $(I, G(I))$ with augmentations, or by manually selecting meaningful pairs (I, J), usually based on matching the categories of the objects contained in the images. However, manual selection undermines the unsupervised nature of these approaches and also limits

correspondences to only those that are manually deemed possible.

Instead, we propose to construct the pairs $\mathcal{P}$ from the images $\mathcal{D}$ automatically, in a process that we call *auto-curation*, based on global image similarity. We use the cls token as a global image descriptor and, given an image I, we construct pairs by randomly sampling from the top-k nearest neighbors images J in the dataset, based on this descriptor. This means that each image is potentially paired with k other images during training, generally yielding many more matching pairs than in the pre-defined case.

Finally, our training scheme also requires to sample points u in the source images. Semantic correspondences are likely only meaningful for foreground objects. Thus, during training, we sample u within the salient region $\Omega_I \subset [0,1]^2$ of the source image I, which we extract using the unsupervised segmentation method of [Y. Wang et al. 2022a].

With this, we can define the overall training objective for the matcher $\hat{v}_\theta$:

$$E(\theta) = \frac{1}{|\mathcal{P}|} \sum_{(I,J)\in\mathcal{P}} \frac{1}{|\Omega_I|} \sum_{u\in\Omega_I} \mathcal{L}(\theta|u,I,J)$$

which combines losses (2.4) and (2.2) as:

$$\mathcal{L}(\theta|u,I,J) = \mathcal{L}_d(\hat{v}_\theta(u;I,J)|u,I,J) + \lambda\mathcal{L}_c(\hat{v}_\theta|u,I,J).$$

2.4 Experiments

To evaluate the effectiveness of our method, we experiment on several benchmarks for semantic correspondences. We compare our automated results with methods trained in an unsupervised and weakly-supervised manner using manually-defined pairs of images of the target classes.

2.4.1 Datasets and Metrics

We perform experiments on four different datasets: PF-Pascal [Ham et al. 2017], Spair-71k [J. Min et al. 2019], CUB-200-2011 (CUB) [Welinder et al. 2010] and Stanford Dogs Extra (SDogs) [Khosla et al. 2011; Biggs et al. 2020].

These datasets cover a wide range of objects and have a different level of difficulty. PF-Pascal contains 1351 image pairs, of which 300 are in the test set, selected from all 20 categories in PASCAL-VOC. The CUB and SDogs datasets contain images for fine-grained recognition. They have sparse keypoint labels, and contain images with varying appearances, poses, and backgrounds, making them particularly challenging. We test on 10,000 random pairs (the same as in [Aygun and Mac Aodha 2022]). Spair-71K is the most challenging dataset of the four, and the only one collected for the task of semantic correspondences. It contains 1800 images, from which 71k image pairs are created, and contains multiple classes, from animals to man-made objects. What makes it particularly challenging are the different poses and appearance of the objects, their small size in some of the images, and the additional objects present in the scenes.

For all datasets, we evaluate using the standard metric for this task: percentage of correct keypoints (PCK). Given a set of ground-truth points $\mathcal{P} = \{p_m\}_{m=1}^{M}$ and predictions $\hat{\mathcal{P}} = \{p_m\}_{m=1}^{M}$, PCK is given by:

$$\text{PCK}(\mathcal{P}, \hat{\mathcal{P}}) = \frac{1}{M} \sum_{m=1}^{M} \mathbb{1}[\|\hat{p}_m - p_m\| \leq \delta].$$

Here, δ is a distance threshold given by $\delta = \alpha \max(H, W)$, where $0 < \alpha < 1$ is a chosen ratio and (H, W) is the image or bounding box size, depending on the dataset convention. Following standard practice, we report α_{img} for PF-Pascal and α_{bbox} for SPair-71K, CUB and SDogs. In all evaluations we use the standard $\alpha = 0.1$ unless stated otherwise.

2.4.2 Implementation Details

We train a single model on 6M images (to keep NN selection tractable) from ImageNet [Jia Deng et al. 2009] and evaluate it on all benchmarks. Our experiments use DINO-pretrained ViT-B [Caron et al. 2021b] as feature encoder. All images are resized to 224×224 and the ViT uses 8×8 patches with stride 8 are used as input. We use the 384-dimensional key features from the 9th layer, similarly to prior work [Amir et al. 2021a; Aygun and Mac Aodha 2022]. The temperature for the contrastive distillation loss $\mathcal{L}_d$ in Eq. (2.2) is set to $\tau_d = 0.1$. During training, we set the temperature for the scoring function in Eq. (2.1) to $\tau_H = 0.05$, and,

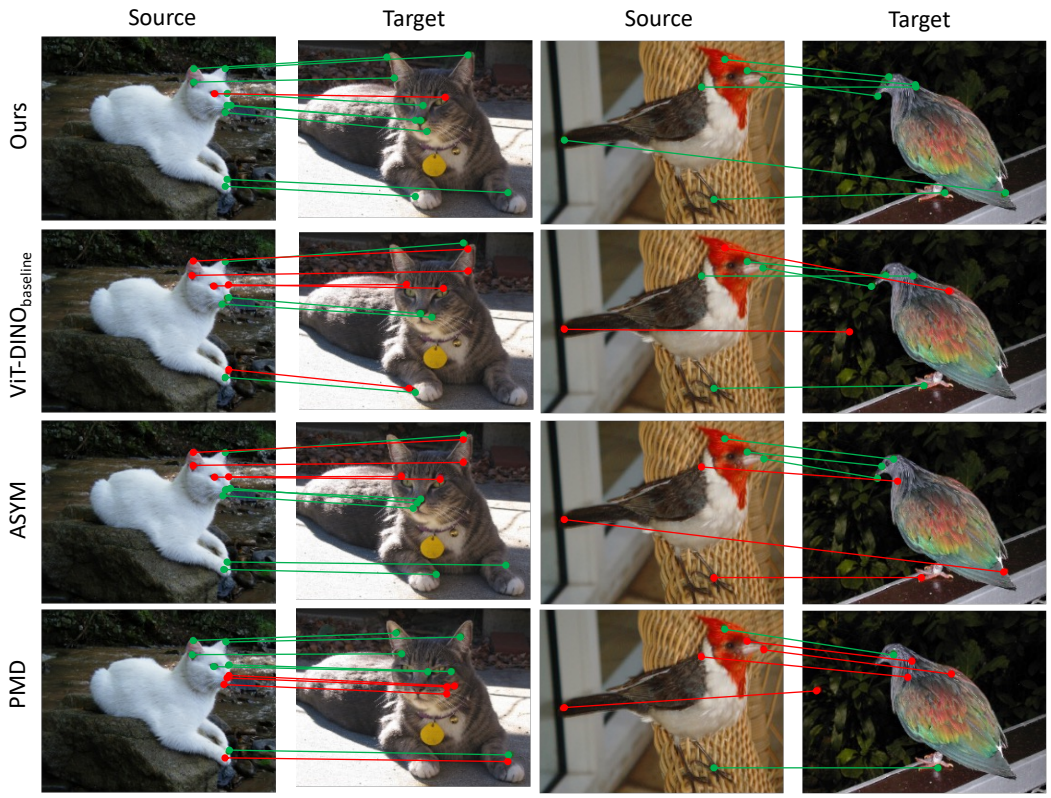


Figure 2.5: **Qualitative Comparison on SPair-71k**. Our method can find good semantic correspondences under strong appearance differences. PMD achieves state-of-the-art performance on the well-aligned PF-Pascal dataset, but struggles for complex image pairs. We provide more qualitative evaluating in the sup. matt.

during testing, we effectively set the temperature to the theoretical limit $\tau_H \to 0$ by using the arg max operator in Eq. (2.3). We set the weight in the expression for the loss $\mathcal{L}$ to $\lambda = 5$.

For the descriptor adapter, we use the transformer decoder architecture of [Jaegle et al. 2021] with 4 transformer layers, setting the number of self-attention heads to 8 and the feed-forward dimensionality to 1024. To all spatial descriptors we concatenate sinusoidal positional embeddings, and to the `cls` tokens we concatenate learnable embeddings. The individual queries do not interact with each other, and during test-time only the locations of the query points are computed. During training, we randomly sample 200 points from the salient region of the source image.

In the cycle consistency loop of Eq. (2.4) we construct the transformation G as follows: (1) we sample a thin-plate spline transformation by randomly jittering a 10×10 grid, (2) we perform a random rescaled crop, and (3) apply color jittering. All further implementation details can be found in the supplementary material.

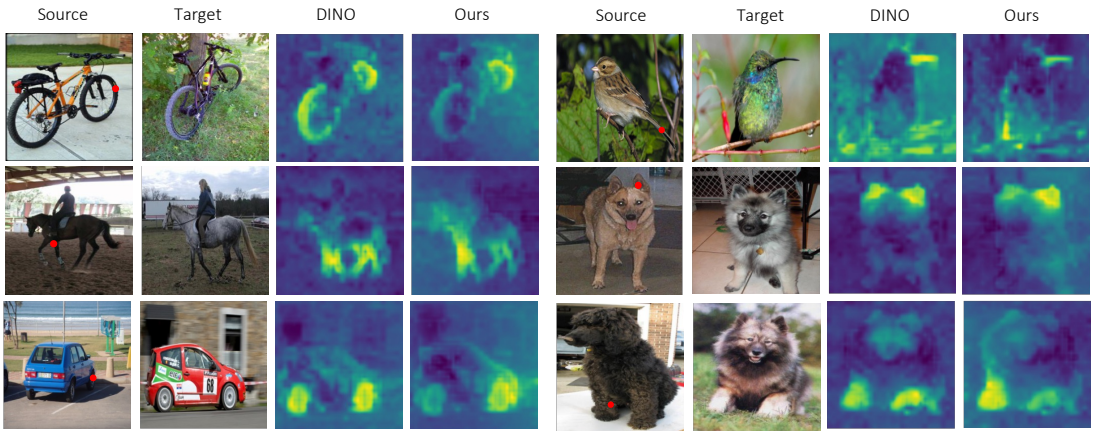


Figure 2.6: **Similarity Heatmaps.** We show the similarity between the target image and (1) ViT-DINO feature sampled at the query location marked in red (2) an adapted feature by our transformer adapter, resulting in less ambiguous matches. The pairs are from PF-Pascal, SPair71k, CUB and SDogs.

Code and models will be released upon acceptance of the paper.

We use a schedule for the size of the image-pair neighborhood k, gradually increasing k and thus the diversity of pairs during training. In particular, we train with $k = 5$ for the first 10k iterations, $k = 100$ for the next 20k iterations, and $k = 500$ thereafter.

2.4.3 Results

Table 2.1 compares our results against the state of the art (weakly and unsupervised methods) on four datasets. We note that all other methods in Table 2.1 are *dataset specific* – they train on images from the same dataset they use for evaluation, whereas we train on one dataset (ImageNet-21k) and evaluate on different datasets at test time.

Our method improves the baseline ViT-DINO features on all datasets. This is in contrast to another recent method [Aygun and Mac Aodha 2022], whose objective for unsupervised correspondences leads to lower performance on some benchmarks when using ViT-DINO.

We report state-of-the-art results in unsupervised semantic correspondences on three out of four datasets, SPair-71K dataset, CUB and SDogs. In SPair-71K and CUB we also outperform weakly-supervised methods which use manually-defined image pairs for all classes.

	Method	DS	Spair-71k	PF-Pascal	CUB	SDogs
W	GANgealing [Peebles et al. 2022]	✓	—	—	57.5	—
	NACongealing$_{\text{ViT-DINO}}$ [Ofri-Amar et al. 2023]	✓	—	—	63.2	—
	WeakAlign $_{\text{res101}}$ [Ignacio Rocco et al. 2018]	✓	21.1	75.8	—	—
	CL$_{\text{ViT-DINO}}$ [Cheng et al. 2021; Aygun and Mac Aodha 2022]	✓	25.8	—	54.1	32.3
	NC-Net$_{\text{res101}}$ [I. Rocco et al. 2018]	✓	26.4	78.9	—	—
	PMD$_{\text{res101}}$ [Xin Li et al. 2021]	✓	26.5	81.2	39.6†	32.2†
	DHPF$_{\text{res101}}$ [J. Min et al. 2020]	✓	28.5	82.1	—	—
	ASYM$_{\text{ViT-DINO}}$ [Aygun and Mac Aodha 2022]	✓	32.9	—	65.2	45.2
	GSF$_{\text{res101}}$ [Jeon et al. 2020]	✓	33.5	84.5	—	—
	LEAD$_{\text{ViT-DINO}}$ [Karmali et al. 2022; Aygun and Mac Aodha 2022]	✓	33.6	—	60.8	42.5
	PWarpC-NC-Net$_{\text{res101}}$ [Truong et al. 2022]	✓	35.3	84.4	—	—
U	PMD$_{\text{res101}}$ [Xin Li et al. 2021]	✓	—	**80.5**	—	—
	CNNGeo$_{\text{res101}}$ [Ignacio Rocco et al. 2017]	✓	18.1	69.5	—	—
	Sem-GLU-Net$_{\text{vgg16}}$ [Truong et al. 2020b; Truong et al. 2022]	✓	16.5	72.5	—	—
	A2Net$_{\text{res101}}$ [P. H. Seo et al. 2018]	✓	20.1	70.8	—	—
	ViT-DINO $_{\text{baseline}}$	✗	34.1	68.3	61.0	42.7
	Ours	✗	**35.9**	73.5	**66.2**	**43.7**

Table 2.1: **Evaluation.** W: weakly supervised. U: unsupervised. DS: dataset specific models that train on images from the dataset used for evaluation. †: evaluation of [Prince et al. 2012] using published weights and code of SPair-71k pre-trained model. Our method improves over prior unsupervised *and* weakly supervised work on three out of four challenging benchmarks. PF-Pascal contains only aligned image-pairs, favoring models with strong geometric regularization that often do not generalize well to the other datasets.

PMD [Xin Li et al. 2021] works better than our approach on the PF-Pascal dataset. PF-Pascal is different from the other benchmarks as it only contains image pairs with lower complexity correspondences: all image pairs are of aligned objects that have the same pose and their relative transformation can be approximated with a simple warp. Models that predict a flow field between the two images (such as PMD) have a clear advantage in this case, but fail when there is a complicated non-rigid transformation of the object between the source and the target. To support this hypothesis, we evaluate PMD's released *weakly* supervised model on CUB and SDogs and indeed see low performance outside of PF-Pascal (see also Fig. 2.5).

Per-class evaluation on PCK. To better understand the semanticity of the learned correspondences, we evaluate per-class PCK on SPair-71k in Table 2.2. We note that our model mainly achieves state-of-the-art on animal classes. We hypothesize a reason for this lies in the way we sample query points during training: randomly from the salient region of the image. For animals, most keypoints of interest are *within* the body of the animal, while for other objects, such as car, bus, TV, the points of interest usually lie at the boundary. Due to imperfections in the unsupervised saliency masks, our method less often samples boundary points

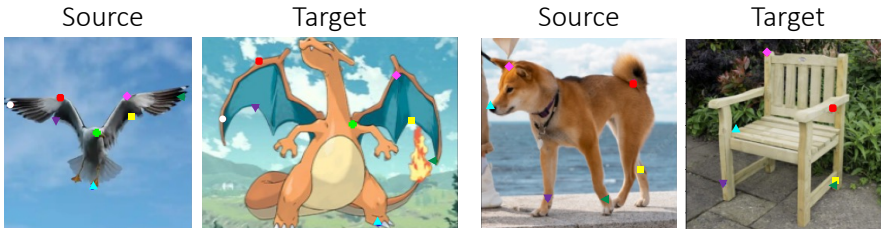


Figure 2.7: **Qualitative evaluation on out-of-distribution data.** Left: there are still semantic similarities that our model manages to find. Right: failure—there are no semantic correspondences.

and thus does not perform as well when the query points are at the boundary of the object at test time.

Sup.	Method																		all
W	WeakAlign [Ignacio Rocco et al. 2018]	22.2	17.6	41.9	15.1	**38.1**	27.4	27.2	31.8	12.8	26.8	22.6	14.2	20.0	22.2	17.9	10.4	32.2	35.1	20.9
	$SFNet_{res101}$ [J. Lee et al. 2019]	26.9	17.2	45.5	14.7	38.0	22.2	16.4	55.3	13.5	33.4	27.5	17.7	20.8	21.1	16.6	15.6	32.2	35.9	26.3
	PMD_{res101} [Xin Li et al. 2021]	26.2	18.5	48.6	15.3	38.0	21.7	17.3	51.6	13.7	34.3	25.4	18.0	20.0	24.9	15.7	16.3	31.4	**38.1**	26.5
	$ASYM_{ViT-DINO}$ [Aygun and Mac Aodha 2022]	38.7	22.6	68.5	21.2	30.7	27.6	25.3	55.9	18.7	44.5	43.9	33.3	20.7	38.6	16.7	31.7	35.2	25.3	32.9
	$LEAD_{ViT-DINO}$ [Karmali et al. 2022]	43.0	24.2	68.6	23.4	31.5	26.2	25.5	57.1	19.5	43.2	42.4	32.0	22.9	41.1	19.4	29.9	34.2	26.2	33.6
U	CNNGeo [Ignacio Rocco et al. 2017]	23.4	16.7	40.2	14.3	36.4	27.7	26.0	32.7	12.7	27.4	22.8	13.7	20.9	21.0	17.5	10.2	30.8	34.1	20.6
	A2Net [P. H. Seo et al. 2018]	22.6	18.5	42.0	16.4	37.9	**30.8**	**26.5**	35.6	13.3	29.6	24.3	16.0	21.6	22.8	20.5	13.5	31.4	36.5	22.3
	$ViT\text{-}DINO_{baseline}$	**44.5**	24.7	68.6	**24.2**	31.7	26.6	26.0	57.3	19.6	44.3	42.9	33.0	**23.4**	40.6	19.7	30.8	34.3	26.7	34.1
	Ours	40.6	**24.9**	**74.0**	18.9	37.2	26.7	22.3	**59.5**	**20.3**	**49.3**	**46.3**	**37.2**	18.5	**41.5**	**21.3**	**35.2**	**43.8**	32.2	**35.9**

Table 2.2: **Per-class evaluation on SPair-71k.** W: weakly supervised. U: unsupervised. Over all classes our method performs best. Additionally, in most cases our method improved over its teacher ViT-DINO. Mechanical objects (trains, cars, and motorbikes) are challenging for our method, potentially due to keypoints often appearing at the boundaries of objects which are underrepresented in our training.

Efffect of weak supervision. To explore the competing effects of using (1) less training data, and (2) manually-defined image pairs (i.e. weak supervision), we train our model on SPair71k, CUB and SDogs (as in [Aygun and Mac Aodha 2022]), using the provided manually-defined pairs and the same hyperparameters as before. Table 2.3 shows that using manually supervised image pairs does improve performance, but only slightly (∼1% relative improvement). Note that, by using manually defined image pairs in these smaller datasets, we observe the compound effect of using fewer, cleaner and more specific pairs. However, given that the difference in performance is small, this validates the robustness and *generalization* of our model, which is also *fully unsupervised*.

Qualitative results. In Fig. 2.1 we show that our model can find good semantic correspondences across different scales (face, full body), and between categories (bird-tiger). We qualitatively compare our method to prior work in Fig. 2.5. We see that our method is robust across pose, scale, and appearance changes. Finally, in Fig. 2.7 we show results on out-of-distribution examples (animation and unrelated

Sup.	Methods	**SPair71k**	**CUB**	**SDogs**
W	$\text{CL}_{\text{ViT-DINO}}$ [Cheng et al. 2021]	25.8	54.1	32.3
	$\text{LEAD}_{\text{ViT-DINO}}$ [Karmali et al. 2022]	33.6	60.8	42.5
	$\text{ASYM}_{\text{ViT-DINO}}$ [Aygun and Mac Aodha 2022]	32.9	65.2	**45.2**
	Ours	**36.4**	**66.8**	44.2
U	$\text{DINO}_{\text{baseline}}$	34.1	61.0	42.7
	Ours	35.9	66.2	43.7

Table 2.3: **Using Weak Supervision.** Weak supervision trades cleanliness of training pairs with limited amount of training data resulting in marginal improvements.

Loss							
$\mathcal{L}_c$	$\mathcal{L}_{dT}$	$\mathcal{L}_{dS}$	Arch.	Mask	**Spair-71k**	**CUB**	**SDogs**
✓	✗	✗	*LGa*	✓	17.5	29.0	23.9
✓	✓	✗	*LGa*	✓	31.7	60.6	42.2
✓	✗	✓	*LGa*	✓	33.0	64.2	42.4
✗	✓	✓	*LGa*	✓	33.9	64.4	42.2
✓	✓	✓	*LLa*	✓	32.3	62.9	41.9
✓	✓	✓	*La*	✓	32.8	63.6	42.7
✓	✓	✓	*LGa*	✗	27.6	57.9	39.0
✓	✓	✓	*LGa*	✓	**35.9**	**66.2**	**43.7**

Table 2.4: **Ablation studies.** Losses: $\mathcal{L}_c$, $\mathcal{L}_{dT}$, and $\mathcal{L}_{dS}$ are the cycle consistency, distillation-from-target, and distillation-from-source losses, respectively. Adaptation: *La*, *LLa*, and *LGa* are Local, Local/Local, and Local/Global adaptation techniques, respectively.

class). Our model can find good correspondences when there are semantically similar object parts in the two images.

In Fig. 2.6 we show heatmaps from cosine similarities between the target images and the ViT-DINO query features or the query features we get from our transformer adapter. We see that the adapter successfully disambiguates between left and right, and front and back. They are also more precise, leading to better localization.

Ablation Study. First, we explore the performance using different losses and architectures in Table 2.4. The losses $\mathcal{L}_{dT}$ and $\mathcal{L}_{dS}$ form the contrastive loss in Eq. (2.2), and they represent sampling negatives from the target and source image, respectively (as in Fig. 2.4). We find that sourcing from both images is important, and the geometric prior of the cycle consistency improves the performance of the imperfect ViT-DINO features. Furthermore, adapting using *LGa*

Schedule	**Spair-71k**	**CUB**	**SDogs**
Random	27.9	53.3	39.6
Top_5 NNs	29.5	58.9	40.3
Top_{100} NNs	29.5	58.6	40.6
Top_{500} NNs	28.6	57.1	40.6
$\text{Top}_{5\to100\to500}$ NNs	**35.9**	**66.2**	**43.7**

Table 2.5: **Pair Selection Strategy.** A curriculum of gradually increasing the sampling neighborhood improves results.

is superior to the other methods. Interestingly, *La* performs better than *LLa* — this confirms our hypothesis that, although this architecture is successful in the supervised regime, it is prone to shortcut learning and poor performance in the unsupervised setting. Finally, sampling points from pseudo masks further boosts results. This is expected, as, without the mask, we force the model to find correspondences in the background, which often do not exist.

Next, we look at the importance of using a schedule when selecting nearest neighbor pairs during training (Table 2.5). Using a random pair results in the lowest performance, as often pairs do not contain meaningful correspondences (e.g. cup and zebra, etc.). Even when we only sample from a small neighborhood (top five NNs), performance already greatly improves. When we increase the neighborhood size without using a schedule, we see no further gains, and in fact, it drops when we increase to 500. We explain this with the fact that while correspondences across large appearance changes are important for the model to pick up, they are difficult to discover without supervision. We solve this with the schedule that gradually increases the difficulty of the pairs.

2.5 Conclusions

We have presented a method to learn image correspondences without supervision. The method starts from a self-supervised ViT feature extractor and improves it by learning an adapter network that, given a set of features form a source image and the target image, steers the source features to better match the target. The method can be trained on a general-purpose image dataset without requiring manually-paired images for supervision. It obtains high-quality matches which

outperform previous unsupervised methods, particularly on the hardest datasets, where matching requires a good degree of semantic abstraction.

Ethics. We use the PF-Pascal, SPair-71k, CUB and Stanford Dogs Extra datasets in a manner compatible with their terms. Some of these images may accidentally contain personal data (faces), but there is no extraction of personal or biometric information in this research. For further details on ethics, data protection, and copyright please see https://www.robots.ox.ac.uk/~vedaldi/research/union/ethics.html.

Acknowledgements. We thank Tim Franzmeyer, Bruno Korbar and Yash Bhalgat for proofreading. A. Shtedritski is supported by EPSRC EP/S024050/1. A. Vedaldi and C. Rupprecht are supported by ERC-CoG UNION 101001212. C. Rupprecht is also partially supported by VisualAI EP/T028572/1.

2.6 Appendix

In this supplementary material, we provide additional details about our approach, experiments and results.

2.6.1 Discussion of the Architecture

Here we further discuss the architecture of our model.

Feature Extractor To evaluate the importance of using ViT-DINO features, we compare a number of different pre-trained self-supervised ViTs in Table 2.6. We find that our method works best with ViT-DINO features, followed closely by ViT-MSN [Assran et al. 2022]. MAE does not seem to learn features with strong semantic correspondence properties, possibly due to its image reconstruction objective.

Fine-tuning the Backbone. We found that keeping the ViT-DINO backbone frozen to be essential for our method to work well. If the pre-trained ViT-DINO is fine-tuned together with the transformer adapter, it returns features Φ that minimize the contrastive loss in Eq. 2 but have lost their semantic power. We

Method	Backbone	**CUB**	**Spair71k**	**PF-Pascal**
MAE [He et al. 2021]	ViT-16	34.6	18.4	54.7
MSN [Assran et al. 2022]	ViT-16	62.1	29.5	64.8
DINO [Caron et al. 2021b]	ViT-16	64.9	–	–
DINO [Caron et al. 2021b]	ViT-8	**66.2**	**36.4**	**73.4**

Table 2.6: **Comparison of different backbones.** We compare the PCK of a model trained on CUB with different self-supervised ViTB-16 backbones (we use ViTS-8 in our all other experiments, but ViT-8 weights were not available for some of the backbones we evaluate here).



Figure 2.8: **Semantic Correspondences.** Given annotations in one frame, our model can find correspondences when the object undergoes significant appearance and pose changes (transformation to a wolf) in the video.



Figure 2.9: **Qualitative Evaluation on PF-Pascal.** We see that PMD manages to find correct correspondences when the source and target images are aligned.

thus keep the teacher frozen in all our experiments. Moreover, keeping the features used in the distillation loss frozen, while fine-tuning the features that go into the transformer adapter, does not lead to any significant gains in performance.

Additional ablations We ablate the values of the temperature parameters τ_H and τ_d on CUB in Table 2.7. We see that performance is highly dependent on both parameters.

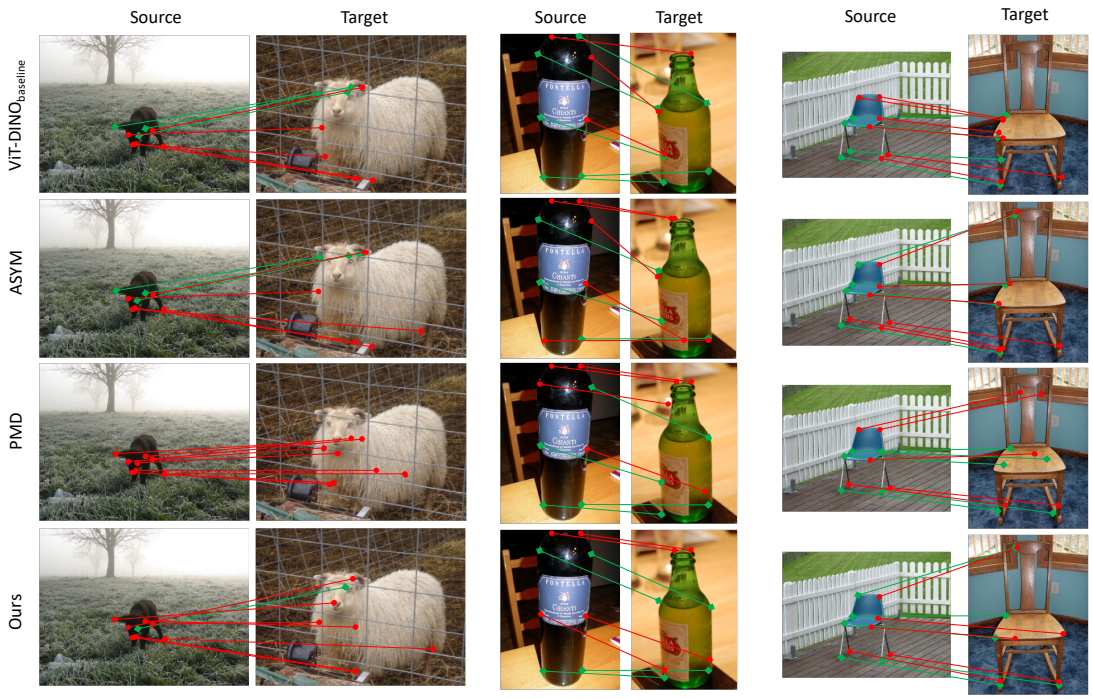


Figure 2.10: **Qualitative Evaluation on SPair-71k.** We see this is a much more difficult dataset than PF-Pascal.

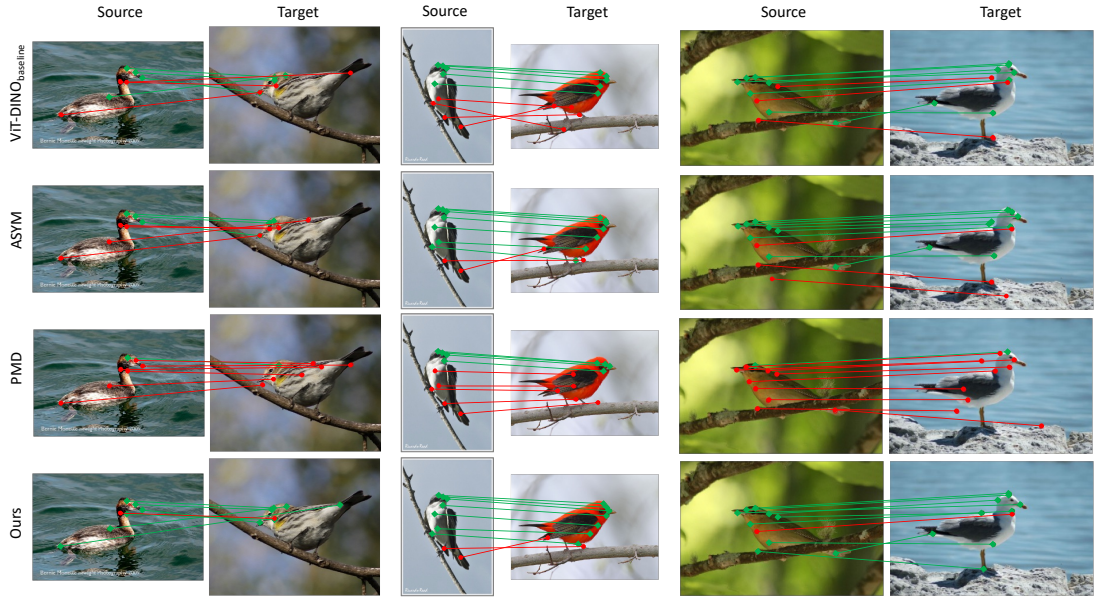


Figure 2.11: **Qualitative Evaluation on CUB.** Our method improves on the baseline features in almost all example points. We see that PMD fails to establish correspondences when there are large pose changes between the source and target images.

2.6.2 Implementation details

In the contrastive distillation loss in Eq. 2, the denominator contains negative features from the source and target image. In practice, we sample 200 features from both images due to memory constraints. We found that the model converges

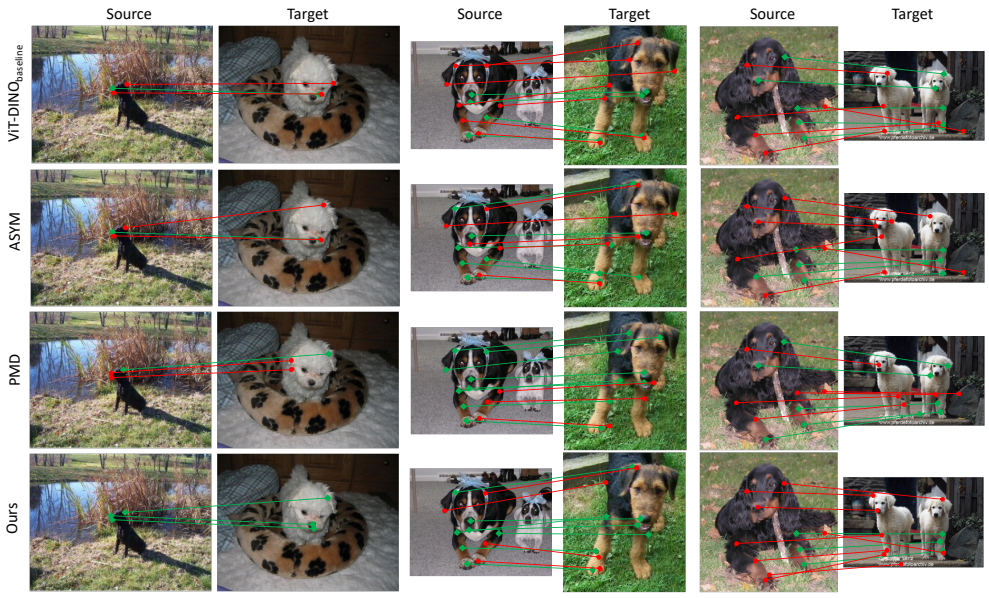


Figure 2.12: **Qualitative Evaluation on SDogs.** Our method generally improves on the baseline features. We emphasize the last column, where the dog on the right is arbitrarily annotated as the correct match.

Value	Ablated param.	
	τ_H	τ_d
1.0	32.4	38.8
0.1	65.3	**66.8**
0.05	**66.8**	66.7
0.01	64.1	66.4

Table 2.7: **Ablation of τ_H and τ_d .** When ablating τ_H, we keep $\tau_d = 0.1$ and when ablating τ_d, we keep $\tau_H = 0.05$. We ablate on CUB.

faster as more negative features are used, but got diminishing returns when using more than 200.

When selecting pseudo pairs in training, we discard images whose salient region is smaller than 20% of the area of the image. We found that many of the unsupervised masks of [Y. Wang et al. 2022a] were very small and not representing any object, and we use this step to filter such masks.

While during training we keep features at their original 28×28 spatial resolution, during test-time in the final matching step we bilinearly upsample the target features to 128×128, and do this for all baselines and comparisons, too. We found this improves matching performance for small objects across the board.

The adapter has 14M parameters and the FLOPs for a single pass are 15B. We

train on a single 48G GPU with batch size 32 for less than 2 days.

2.6.3 Qualitative Evaluations

Our model can track keypoints on an object as it undergoes significant appearance changes, as shown in Fig. 2.8.

In Figs. 2.9 to 2.12 we qualitatively compare our method to the baseline ViT-DINO features and the methods from [Aygun and Mac Aodha 2022; Xin Li et al. 2021] on random examples from PF-Pascal, Spair-71k, CUB, and SDogs. We use green squares for correct matches and red circles for incorrect ones.

We see that as discussed in the paper, PMD performs very well when the transformations between the objects can be approximated by a simple warp, as in the PF-Pascal dataset in Fig. 2.9. In the evaluations of PMD on Spair-71k, CUB and Sdogs, we use their weakly-supervised SPair-71k pre-trained model. We see that in all these cases the model fails to establish semantic correspondences.

Finally, we note a problem of the SDogs dataset that can be seen in the last column of Fig. 2.12. In a large number of the pairs, there is more than one dog in the target image. This leads to ambiguous annotated keypoints, where there should be more than one correct correspondence in the target image. Upon closer inspection, our model (and other models as well) predicts correct keypoints, but not on the arbitrarily selected dog with the ground-truth correspondences.

Chapter 3

SHIC: Shape-Image Correspondences with No Keypoint Supervision

The paper was presented at ECCV 2024.

SHIC: Shape-Image Correspondences with no Keypoint Supervision

Aleksandar Shtedritski, Christian Rupprecht, Andrea Vedaldi

Visual Geometry Group, University of Oxford

{suny, chrisr, vedaldi}@robots.ox.ac.uk

robots.ox.ac.uk/vgg/research/shic/

September 29, 2025



Figure 3.1: **Unsupervised canonical maps.** We show predictions from our *fully unsupervised* method SHIC, which finds correspondences between a rigid 3D template and a natural image. Correspondences are color-coded by assigning a distinct color to each template surface point. Our approach is highly data-efficient; the elephant, T-Rex, and Appa models above are trained on only 2800, 480, and 180 images, respectively.

Abstract

Canonical surface mapping generalizes keypoint detection by assigning each pixel of an object to a corresponding point in a 3D template. Popularised by DensePose for the analysis of humans, authors have since attempted to apply the concept to more categories, but with limited success due to the high cost of manual supervision. In this work, we introduce SHIC, a method to learn canonical maps without manual supervision which achieves *better* results than supervised methods for most categories. Our idea is to leverage foundation computer vision models such as DINO

and Stable Diffusion that are open-ended and thus possess excellent priors over natural categories. SHIC reduces the problem of estimating image-to-template correspondences to predicting image-to-image correspondences using features from the foundation models. The reduction works by matching images of the object to non-photorealistic renders of the template, which emulates the process of collecting manual annotations for this task. These correspondences are then used to supervise high-quality canonical maps for any object of interest. We also show that image generators can further improve the realism of the template views, which provide an additional source of supervision for the model.

3.1 Introduction

Correspondences play an important role in computer vision, with applications to pose estimation, 3D reconstruction, retrieval, image and video editing and many more. In this paper, we consider the problem of learning dense keypoints for any given type of objects without manual supervision. Keypoints identify common object parts, putting them in correspondence, and providing a key abstraction in the analysis of the objects' geometry and pose. While keypoints are usually small in number, dense keypoints [Rıza Alp Güler et al. 2018] are a generalization that considers a continuous family of keypoints indexed by the surface of a 3D template of the object. Dense keypoints provide more nuanced information than sparse ones and have found numerous applications in computer vision and computer graphics.

Despite their utility, learning keypoints, especially dense ones, remains labour-intensive due to the need to collect suitable manual annotations. Because of this, most keypoint detectors are limited to specific object classes of importance in applications, such as humans [Rıza Alp Güler et al. 2018; Rempe et al. 2021; Hongwen Zhang et al. 2021; Kreiss et al. 2019]. Methods that generalize to more categories either have limited performance [N. Kulkarni et al. 2019; N. Kulkarni et al. 2020], or require a significant amount of manual annotations for each class [Neverova et al. 2020; Neverova et al. 2021]. They cannot scale to learning (dense) keypoints for the vast majority of object types in existence.

In contrast, foundation models such as DINO [Caron et al. 2021a], CLIP [Radford et al. 2021], GPT-4 [*ChatGPT* n.d.], DALL-E [Ramesh et al. 2022], and Stable

Diffusion [Rombach et al. 2021] are trained from billions of Internet images and videos with almost no constraints on the type of content observed. While these models do not provide explicit information about the geometry of objects, we hypothesise that they may do so *implicitly* and may thus be harnessed to generalize geometric understanding to more object types.

In this paper, we test this hypothesis by utilizing off-the-shelf foundation models to learn automatically high-quality dense keypoints. Given a single template mesh for an object class (e.g., a horse or a T-Rex) to define the index set for the keypoints, and as few as 1,000 masked example images of the given class, we learn a high-quality image-to-template mapping.

Our method builds on recent advances in self-supervised image-to-image matching algorithms which, by using features from DINO [Caron et al. 2021a] and the Stable Diffusion encoder [Rombach et al. 2021], can generalize surprisingly well across images of different modalities or styles, such as natural images, animations or abstract paintings. Our idea is to reduce the problem of matching images to the 3D template to the one of matching images to *rendered views of the template*. Namely, we render a view of the 3D template and, given a query location in the source image, we find the corresponding vertex as a visual match on the rendered images. The template renders are *not* photorealistic, so the matching process emulates the process of manually annotating dense keypoints in prior works [Rıza Alp Güler et al. 2018; Neverova et al. 2020]. We contribute several ideas to robustly pool information collected from different renders of the template, including accounting for visibility.

The approach we have described so far is training-free, as it uses only off-the-shelf components, but it is slow and the resulting correspondences lack spatial smoothness as they are established greedily. Our second step is thus to use these initial correspondences to supervise a more traditional dense keypoint detector in the form of a canonical surface map [Thewlis et al. 2017a; Rıza Alp Güler et al. 2018; N. Kulkarni et al. 2019]. We utilize the Canonical Surface Embedding (CSE) representation of [Neverova et al. 2021], which was designed to learn a mapping for several proximal object classes together (e.g., cow, dog and horse), and can also efficiently represent image-to-template and image-to-image mappings by learning cross-modal embeddings. The most important result is that we can *outperform*

the original manually-supervised model of [Neverova et al. 2021] on their animal classes *without* using any supervision. This means that we can also learn maps for entirely new classes, such as T-Rex or Appa (a flying bison from a TV show), essentially at no cost (Fig. 3.1).

Finally, we note a further use of foundation models for our application: the generation of photorealistic synthetic images of the object. In particular, we show that a version of Stable Diffusion conditioned on depth can be used to texture the 3D images of the template, significantly narrowing the synthetic-to-real gap. These images are good enough to be used to supervise the dense pose map *directly*, with full synthetic supervision. We show that, while this is no substitute for utilizing real images as described above, it does improve the final performance further.

3.2 Related work

Unsupervised image-to-image correspondences. Many authors have sought to establish correspondences between images without manual supervision to address the cost of obtaining labels for this task. Early methods generate training data by applying synthetic warps to images [J. Chen et al. 2019; Ignacio Rocco et al. 2017; Truong et al. 2020b; Melekhov et al. 2019; Truong et al. 2020a], or use cycle consistency losses [Jeon et al. 2018; Truong et al. 2021; Truong et al. 2022; Shtedritski et al. 2023a]. GANs have also been used to supervise dense visual alignment [Peebles et al. 2022]. Recent advances in self-supervised representation learning [Caron et al. 2021a; Oquab et al. 2023] and generative modelling [Rombach et al. 2021] have boosted the quality of unsupervised semantic correspondences significantly. For instance, [Amir et al. 2021a] establish correspondences by seeking matches between DINO features, and [G. Luo et al. 2023a; Xinghui Li et al. 2023; Hedlin et al. 2023; L. Tang et al. 2023; Junyi Zhang et al. 2023a] use Stable Diffusion instead. Similarly, [Dutt et al. 2024; Morreale et al. 2024] use diffusion features to find mesh-to-mesh correspondences. [Junyi Zhang et al. 2023a] show that DINO and Stable Diffusion features are complementary, the first capturing precise but sparse correspondences and the second the general layout, and propose to combine them. In our work, we use the SD-DINO [Junyi Zhang et al. 2023a] features for matching images.

Animal pose estimation. While most works on pose estimation focus on humans [Felzenszwalb et al. 2008; Newell et al. 2016; Cao et al. 2017; S. Wei et al. 2016; Rıza Alp Güler et al. 2018], several authors have attempted to estimate the pose of animals by detecting [N. Zhang et al. 2014], matching [Kanazawa et al. 2016] or reconstructing [S. Wu et al. 2023; S. Wu et al. 2021] them, or predicting the parameters of parametric models [Zuffi et al. 2017; Zuffi et al. 2018; Zuffi et al. 2019]. However, these methods do not scale well as they need annotations for each type of animal considered. Our method is most similar to [N. Kulkarni et al. 2020; N. Kulkarni et al. 2019] in that we only require a template shape and a collection of images to learn image-to-shape correspondences. However, our method achieves much better performance, while still using fewer images for training.

Image-to-template correspondences. Finding correspondences between images and a 3D template is useful for understanding the geometry of deformable objects, with several applications. For instance, it is used in biology to study the behaviour of animals [Pereira et al. 2018; Waldmann et al. 2023]. Most prior works focus on humans [Rıza Alp Güler et al. 2018; Rempe et al. 2021; Hongwen Zhang et al. 2021; Kreiss et al. 2019] due to the availability of large-scale datasets of densely annotated image-template pairs, such as DensePose-COCO [Rıza Alp Güler et al. 2018]. Similar datasets exist for animals, such as DensePose-LVIS [Neverova et al. 2021], but are much smaller and still only cover a handful of animal classes. To learn image-to-shape correspondences, [N. Kulkarni et al. 2020; N. Kulkarni et al. 2019] parametrise a 3D shape as a 2D uv map, and use cycle consistency to try to learn correspondences automatically, whereas [N. Kulkarni et al. 2020] also learn to predict articulation. Similarly to [Neverova et al. 2021], we use the CSE representation for learning the correspondences. However, differently from [Neverova et al. 2021], our method does not rely on any human-annotated data.

3.3 Method

In this section, we describe SHIC, our method for learning dense keypoints without manual supervision. First, in Section 3.3.1 we recall the notion of dense keypoints, canonical surfaces and canonical surface maps. Then, in Section 3.3.2 we discuss using self-supervised features to establish dense semantic correspondences between

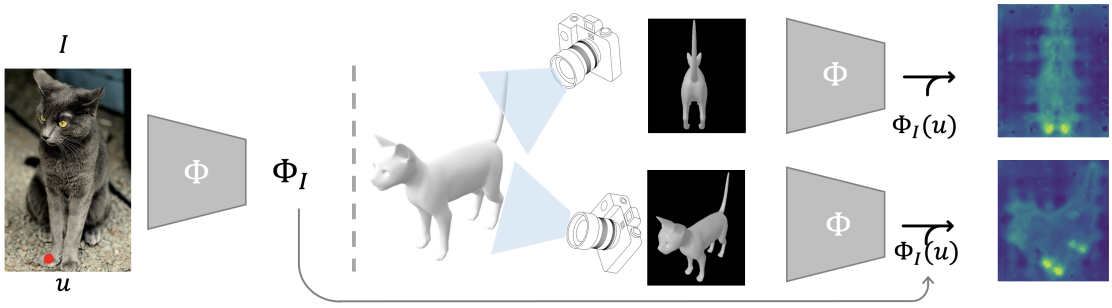


Figure 3.2: **Image-to-template correspondences using 2D renderings.** Using an unsupervised semantic correspondence method, we can find correspondences between an image of an object and a rendering of its 3D template. Here we show the similarity heatmap from the source location (annotated in red) to all pixel locations in the target image using SD-DINO [Junyi Zhang et al. 2023a].

pairs of images, lift those to dense keypoints in Section 3.3.3, and use the latter to supervise a canonical map in Section 3.3.4. Finally, in Section 3.3.5, we show how an image generator can produce realistic views of the template, which further improves results.

3.3.1 Canonical surface maps

Let $I \in \mathbb{R}^{3\times\Omega}$ be an image supported by the grid $\Omega = \{1,\dots,H\}\times\{1,\dots,W\}$. The image contains an object of a given type, such as a cat, and the goal is to assign an identity to each pixel $u \in U_I$ of the object, where $U_I \subset \Omega$ is the object mask in image I. The identification is carried out by a mapping $f_I : U_I \to M$ that assigns each pixel u to a corresponding index $f_I(u)$ in a set M. The set $M \subset \mathbb{R}^2$ is a 2D surface embedded in $\mathbb{R}^3$, and is interpreted as a (fixed and rigid) 3D template of the object. The template M is also called a *canonical surface* and the function f_I a *canonical surface map*. The same canonical surface M is shared by all objects of that category. In this way, by mapping two images I and J to the same template, one can also infer a mapping between the images.

In practice, we approximate the surface M by a mesh supported by a finite set of K vertices $V = \{x_1,\dots,x_K\} \subset M$ and triangular faces F. Hence the canonical map is a function $f_I : \Omega \to V \subset M$. This slightly simplifies the formulation as both index sets Ω and V are finite. We also note that the value $f_I(u)$ is undefined if pixel $u \in \Omega - U_I$ does not belong to the object.

In prior works, learning the canonical map f often requires hundreds of thousands of manually specified image-to-template correspondences. In the next sections, we

will show how to learn this mapping *automatically* instead.

3.3.2 Unsupervised image-to-image correspondences

In order to learn the canonical map f automatically, we start by establishing correspondences between pairs of images I and J in an unsupervised fashion. We do so by first computing D-dimensional dense features $\Phi \in \mathbb{R}^{D\times\Omega}$ using a pre-trained network. Then, we associate each query location u in the source image I to the location v_u in the target image J with the most similar feature vector based on the cosine similarity, i.e.

$$v_u = \arg\max_{v\in\Omega} S_{IJ}(u,v) \quad \text{where} \quad S_{IJ}(u,v) = \frac{\Phi_I(u)\cdot\Phi_J(v)}{\|\Phi_I(u)\|_2\,\|\Phi_J(v)\|_2}.$$

The quality of the correspondences depends on the quality of the feature extractor Φ. In particular, by using the unsupervised features by [Junyi Zhang et al. 2023a], it is possible to establish good correspondences between a (real) image I of the object and a *rendering* of the 3D template M.

This is illustrated in Fig. 3.2, where we show the cosine similarity heatmaps between a feature at a query location u of in the source image I and all locations in several 3D renders of the template. While the correspondences correctly identify the type of body part (paw), two problems are apparent: (i) there is left-right ambiguity, which is common for unsupervised semantic correspondence methods [Junyi Zhang et al. 2023b], (ii) when the correct match is not visible (as on the top of Fig. 3.2, where only the back paws are visible), the correspondence will always be wrong. In the next section, we lift these image-based correspondences into correspondences with the template M, which also alleviates these issues.

3.3.3 Unsupervised image-to-template correspondences

Given the source image I and a pixel u, we now consider the problem of finding the vertex x_k in the template M that best represents it. In order to do so, we develop a similarity measure between pixels and vertices, utilizing the image-to-image similarity metric of Section 3.3.2. We first generate N different views $J_i = \text{Rend}(M, c_i)$, $i = 1,\dots,N$, of the canonical surface M by rendering it from viewpoints c_i (camera parameters). For each view J_i, we project each vertex x_k to

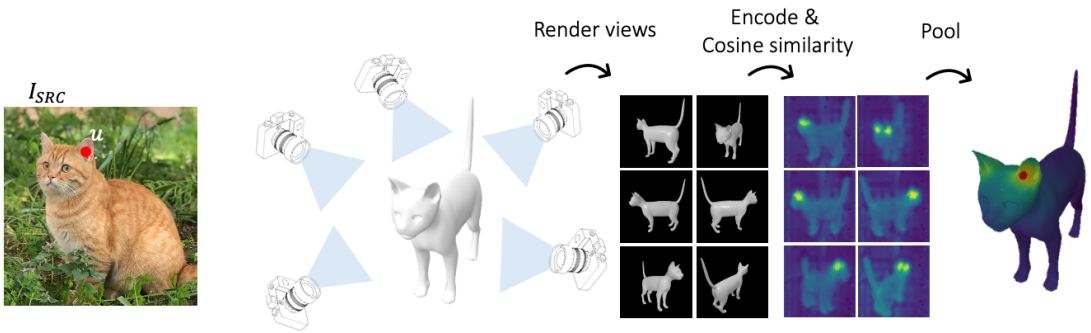


Figure 3.3: **Zero-shot image-to-template correspondences.** From left to right: an image I with a selected pixel u; several views J_i of the synthetic template; corresponding renderings and similarities $S_{IJ_i}(u, v)$ as functions of the target locations $v \in \Omega$; the final similarities $\Sigma_I(u)$ visualized as a heatmap on top of the canonical surface M. The maximizer of the latter (red dot) identifies the vertex x_k that best corresponds to the selected pixel u in the source image I (i.e. base of the left ear of the cat).

its closest location in the mask U_{J_i}, defining

$$v_i(k) = \arg\min_{v \in U_{J_i}} \|v - \pi(x_k, c_i)\|, \tag{3.1}$$

where $\pi(x_k, c_i)$ is the camera projection function. We also denote by $V_i \subset V$ the subset of vertices x_k that are visible in view J_i.

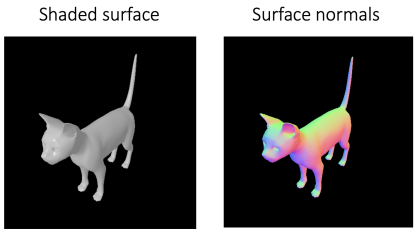
Given this notation, we can define a new score Σ measuring the compatibility between each location u in the source image I and each vertex $x_k \in V$ in the canonical surface, and corresponding matches $\tilde{x}$, as follows:

$$\Sigma_I(u, x_k) = \operatorname*{pool}_{i: x_k \in V_i} S_{IJ_i}(u, v_i(k)), \quad \tilde{x}(u, I) = \arg\max_{x_k \in V} \Sigma_I(u, x_k). \tag{3.2}$$

The goal of the pooling operator is to assess the compatibility between pixel u and vertex x_k into a single score that consolidates the information collected from the different viewpoints c_i. Note that only the views where the vertex is visible are pooled. In practice, we set the pooling operator to average or max pooling.

Illustration. Figure 3.3 illustrates the similarity maps S_{IJ_i} between the source image I and various views J_i of the rendered 3D object, as well as the result Σ_I of mapping and pooling them on the canonical surface itself. We see that the correct semantic parts on the shape are identified (ears), and the base of the left ear is selected as the most similar to the query u.

Template realism and rendering function. The template M captures the typical shape of the object. Mathematically, its main purpose is to define the *topology* of the object's surface, but the latter is usually topologically equivalent to a sphere [Thewlis et al. 2017a]. In dense pose [Neverova et al. 2020] there are two reasons for not using a sphere. The first is that the *metric* of the template surface can be used to regularize correspondences (e.g., by capturing an approximate notion of how far apart physical points are). The second is that *renders* of the template are given to human annotators to establish correspondence with the template. Our method can be seen as automatizing the annotation step. Just like for manual annotation, it does *not* require a photorealistic rendition of the template. However, it does not mean that *all renditions are equally good* from the viewpoint of the matching network. Inspired by previous work on image generation [R. Chen et al. 2023], we find that rendering a *normal map* of the 3D template results in better matches than rendering a shaded version of the same (see the embedded figure). We discuss realistic rendering in Section 3.3.5.



3.3.4 Unsupervised canonical surface maps

Here, we show how to learn the canonical surface map f from the image-to-template correspondences constructed in Section 3.3.3. An overview is in Fig. 3.4.

Continuous Surface Embeddings. Following [Neverova et al. 2020; Neverova et al. 2021], we represent the map f via Continuous Surface Embeddings (CSEs). CSE assign embedding vectors $e_I(u), e(x_k) \in \mathbb{R}^D$ to each image pixel u and each mesh vertex x_k so that the correspondences are defined by maximizing their similarity:

$$f_I(u) = \arg\max_{x_k \in V} p(x_k|u, I), \quad \text{where} \quad p(x_k|u, I) = \frac{\exp(\langle e_I(u), e(x_k)\rangle)}{\sum_{t=1}^{K} \exp(\langle e_I(u), e(x_t)\rangle)}. \quad (3.3)$$

Learning the CSE model thus amounts to learning the vertex embeddings $e(x_k)$ as well as a corresponding dense feature extractor e_I.

The vertex embeddings are optimized directly as there is a single template mesh.

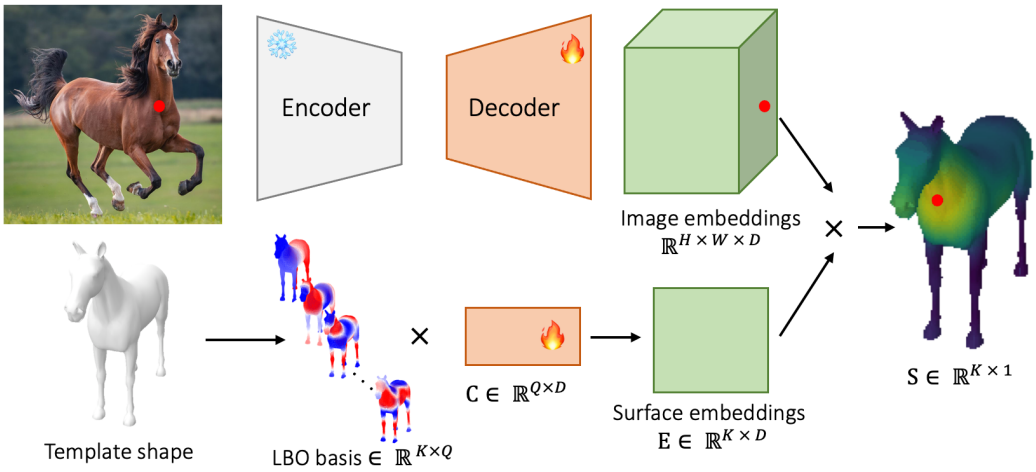


Figure 3.4: **CSE dense pose predictor.** We jointly train a deep network Φ and a matrix C, that transforms LBO eigenvectors to a shared D-dimensional space. We use pseudo-ground truth, obtained as described in Section 3.3.3 for supervision. The image encoder is a frozen pre-trained DINO ViT, and the decoder we learn is a CNN.

However, due to the large number of vertices, they are not assumed to be independent but to form a smooth (vector) function over the mesh surface. This way, the number of parameters required to express them can be reduced significantly. Collectively, all embeddings $e(x_k)$, $k = 1, \ldots, K$, form an embedding matrix $E \in \mathbb{R}^{K\times D}$. The latter is decomposed as the product $E = UC$ where $U \in \mathbb{R}^{K\times Q}$ is a smooth and compact functional basis (akin to Fourier components defined on the mesh) such that $Q \ll K$. Following [Neverova et al. 2020; Neverova et al. 2021], we use the lowest eigenvectors of the Laplace-Beltrami operator (LBO) of the mesh M to form U. The only learnable parameters are $C \in \mathbb{R}^{Q\times D}$, which are few.

The other component is the feature extractor $e_I(u)$. For this, we encode the source image $I \in \mathbb{R}^{3\times H\times W}$ with a frozen self-supervised encoder (DINO [Oquab et al. 2023]), before decoding it with a CNN back to the original resolution to the required feature tensor $e_I \in \mathbb{R}^{D\times H\times W}$.

Training formulation. We train our model with several losses. The first one simply uses the pseudo-ground-truth correspondences $\tilde{x}(u, I)$ of Eq. (3.2) in Eq. (3.3). We pose this as a classification problem, where given a query location u in image I is matched probabilistically to the pseudo-ground-truth $\tilde{x}(u, I)$ using the cross-entropy loss: $\mathcal{L}_{\text{pseudo}}(I) = -\frac{1}{|U_I|}\sum_{u\in U_I} \log p(\tilde{x}(u, I)|u, I)$. Additionally, we use the

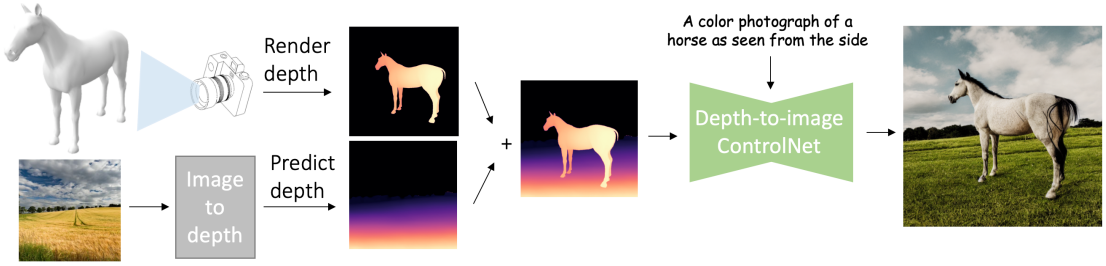


Figure 3.5: **Realistic rendering of the template.** We create synthetic data for pixel-vertex correspondences by generating photorealistic images from depth renders. The corresponding vertices we obtain from the projections of vertices on the image.

distance-aware loss of [Neverova et al. 2021]:

$$\mathcal{L}_{\text{dist}}(I) = -\frac{1}{|U_I|}\sum_{u\in U_I}\sum_{x\in V} d(x,\tilde{x})p(x|u,I),$$

where $d(x,\tilde{x})$ is the geodesic distance between the vertex x and the pseudo ground-truth $\tilde{x}$, which discourages placing probability mass far from $\tilde{x}$.

As noted in Section 3.3.3, there is some ambiguity because of the symmetry of most animals, where the matches are confused between left and right. To reduce this ambiguity, we use a cycle consistency loss [Neverova et al. 2020], where given a starting location u, we match it to a vertex x_k in the mesh, and then matching that back to the image, results in the probability $p(v|u,I) = \sum_{x_k\in V} p(v|x_k,I)p(x_k|u,I)$ of landing to a location v. Here $p(v|x_k,I)$ is the same as $p(x_k|v,I)$ from Eq. (3.3) up to renormalization. We close the image-shape-image cycle and supervise using $\mathcal{L}_{\text{cyc}}(I) = \sum_{u\in U_I}\sum_{v\in U_I}\|u-v\|p(v|u)$.

To further reduce the left-right ambiguity, we assume that the template V has a bilateral symmetry (true for most categories). Then, for each vertex $x \in V$, let $x_F \in V$ be its symmetric one (for meshes which are not exactly symmetric, we let x_F be the closest approximation to the symmetric version of x). Given an image I and a pixel u, denote by I_F and u_F their horizontal flips. Suppose that u is the pixel that corresponds to vertex x in image I. Then one can show [Thewlis et al. 2018] that pixel u_F must correspond to vertex x_F in image I_F, leading to the loss: $\mathcal{L}_{\text{eq}}(I) = \frac{1}{|U_I|}\sum_{u\in U_I}\sum_{x\in V}|p(x|u,I) - p(x_F|u_F,I_F)|$.

3.3.5 Increasing the realism of synthetic data

The renders J_i of the 3D template M can be used to supervise the canonical map f directly because we *know* the 2D location $v_i(k)$ of each vertex x_k in image J_i based on Eq. (3.1). With this, we can write the loss:

$$\mathcal{L}_{\text{syn}}(J_i, v_i) = -\frac{1}{K}\sum_{k=1}^{K}\sum_{i:V_k\in i}\log p(x_k|v_i(k), J_i) \tag{3.4}$$

The very limited diversity and realism of the renders J_i makes this loss uninteresting, but, as shown in Fig. 3.5, we can use a powerful image generator to significantly augment the realism of such renders.

To do this, we first render a depth image of the template M from a random viewpoint c. We also sample a random background image and predict its depth using [Lihe Yang et al. 2024], blend the foreground and background depth images, and use the depth-to-image ControlNet [Lvmin Zhang et al. 2023] of [Lihe Yang et al. 2024] to generate photorealistic image J_i of the template. We prompt the depth-to-image model (i) using the object's class name, e.g., "horse", and (ii) specifying the viewpoint ("front", "side", or "back"), which we heuristically obtain from the camera location w.r.t. the 3D template M.

The results are photo-realistic renders J_i of the template. Note that the appearance of different renders is not consistent, but this is a feature rather than an issue in our case because we need to learn an image-to-template map, which is invariant to details of the appearance. The main limitation is that the template is fixed, so there is no diversity in terms of pose and 3D shape. Hence, we expect loss Eq. (3.4) to be complementary rather than substitutive of the one above.

3.3.6 Learning formulation

Given a dataset $\mathcal{D}$ of masked training images of the object, our loss is:

$$\mathcal{L} = \frac{1}{|\mathcal{D}|}\sum_{I\in\mathcal{D}}(\alpha\mathcal{L}_{\text{pseudo}}(I) + \beta\mathcal{L}_{\text{cyc}}(I) + \gamma\mathcal{L}_{\text{dist}}(I) + \delta\mathcal{L}_{\text{eq}}(I)) + \zeta\sum_{i=1}^{N}\mathcal{L}_{\text{syn}}(J_i, v_i),$$

where $\alpha, \beta, \gamma, \delta$ and ζ are coefficients set empirically.

Method	Supervision	Horse	Sheep	Bear	Zebra	Cow	Elephant	Giraffe	**Average**
CSE [Neverova et al. 2021]	S	24.1	32.0	35.7	24.9	25.4	26.1	**18.0**	26.6
Zero-shot$_{\text{SD-DINO}}$	U	37.2	41.4	48.2	32.0	32.3	36.0	26.3	36.2
SHIC (Ours)	U	**23.3**	**30.3**	**33.0**	**23.3**	**22.7**	**23.9**	18.1	**24.9**

Table 3.1: **Evaluation on DensePose-LVIS.** We compare the supervised (S) method of [Neverova et al. 2021] to our unsupervised method (U) and our adaptation of SD-DINO to DensePose described in Section 3.3.3. We evaluate [Neverova et al. 2021] using their published weights. We measure geodesic error (*lower is better*).

Method	Supervision	Cow	Sheep	Horse	**Average**
Rigid-CSM [N. Kulkarni et al. 2019]	S	28.5	31.5	42.1	34.0
A-CSM [N. Kulkarni et al. 2020]	S	29.2	39.0	44.6	37.6
CSE [Neverova et al. 2021]	S	51.5	46.3	59.2	52.3
Rigid-CSM [N. Kulkarni et al. 2019]	U	26.3	24.7	31.2	27.4
A-CSM [N. Kulkarni et al. 2020]	U	26.3	28.6	32.9	29.3
SHIC (Ours)$_{\text{im2im}}$	U	69.1	55.9	58.7	61.2
SHIC (Ours)$_{\text{im2m2im}}$	U	**73.5**	**73.5**	**63.1**	**70.0**

Table 3.2: **PCK-Transfer on PF-Pascal.** We compare against prior work on image-to-image semantic correspondences. We predict image-to-image correspondences either by directly predicting the correspondences or by performing image-to-vertex-to-image matching. We use the reported numbers from [Neverova et al. 2021; N. Kulkarni et al. 2020; N. Kulkarni et al. 2019], and evaluate using PCK-0.1.

3.4 Experiments

We evaluate our method for learning canonical maps automatically, without manual keypoints supervision, against unsupervised and supervised prior work.

3.4.1 Implementation details

To obtain the image-to-shape similarities Σ, we use $N = 72$ renderings of the template shape (using the surface normal style), render the surface normals, and compute the image-to-image similarities S using the features from [Junyi Zhang et al. 2023a]. For each rendered image, we automatically get the pixel-to-vertex matches from the camera projection function. Finally, we aggregate the similarities across all views using `max` pooling. During training, we randomly select 100 foreground points from each image and their corresponding vertices from Σ. Following [Neverova et al. 2021], we use the lowest $Q = 64$ eigenvectors of the LBO of the template mesh and use dimensionality $D = 16$ for the joint image-shape embedding space. For the image encoder Φ, we use a frozen DINO-v2 [Oquab et al. 2023] backbone, followed by a decoder consisting of 5 convolutional layers. We did

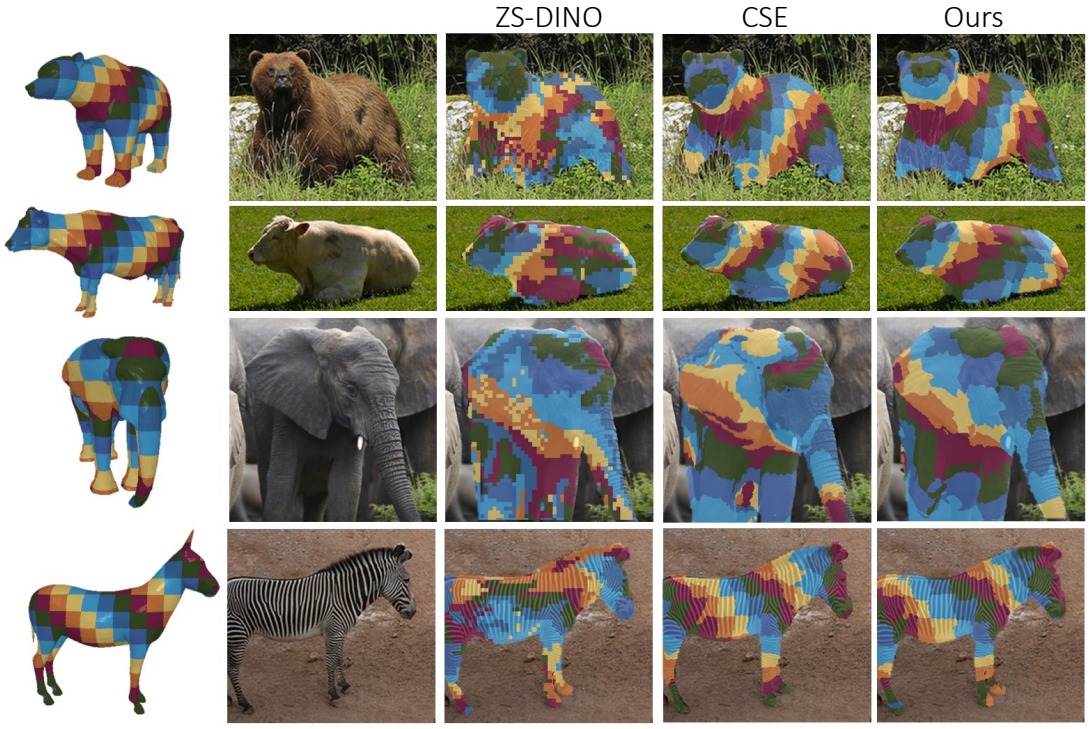


Figure 3.6: **Mapping a textured mesh.** We map a textured mesh over the image using the predicted dense correspondences.

not do any tuning of the parameters in the final formulation of the loss. We used values for the loss hyper-parameters that make the losses roughly of similar scale, namely $\alpha = 0.1, \beta = 0.002, \gamma = 0.002, \delta = 0.001, \zeta = 0.1$. All models and code will be released upon acceptance of the paper.

3.4.2 Training data

For most of our evaluations, we consider the DensePose-LVIS dataset [Neverova et al. 2021], which applies DensePose to a variety of animal classes. Of those, we consider the horse, sheep, bear, zebra, cow, elephant and giraffe classes[1]. The DensePose-LVIS data contains a total of 6k images of these categories, as well as a reference 3D template for each category. Knowledge of the 3D template is necessary to interpret the annotations in the dataset, as well as to compute the geodesic distances required for evaluation. Every animal has up to three manually annotated pixel-template correspondences to the corresponding template. We use these annotations only for evaluation. We only train our models on cropped animals from DensePose-LVIS [Neverova et al. 2021]. The number of instances for each

[1]For the classes cat and dog, we could not obtain the 3D templates from [Neverova et al. 2021] and could therefore not use the annotated image-to-template correspondences for evaluation.

Ablation		DensePose-LVIS
	Ours	**24.9**
Pooling	`mean` (instead of `max`)	26.6
Data	w/o Synth data	25.6
	w/o LVIS	33.8
	w/o pseudo GT	65.8

Table 3.3: **Data and other ablations.** First, we ablate the pooling function used to construct Σ, and evaluate `mean` instead of `max` pooling. Next, we compare the data we use — we remove (1) synthetic data, (2) natural images, (3) the pseudo ground truth for natural images. In (3), we still use natural images for $\mathcal{L}_{\text{eq}}$ and $\mathcal{L}_{\text{cyc}}$, but do not use the pseudo-GT for $\mathcal{L}_{\text{pseudo}}$ and $\mathcal{L}_{\text{dist}}$.

Ablation		DensePose-LVIS
	Ours	**24.9**
Losses	w/o $\mathcal{L}_{\text{pseudo}}$	27.3
	w/o $\mathcal{L}_{\text{dist}}$	25.8
	w/o $\mathcal{L}_{\text{eq}}$	25.6
	w/o $\mathcal{L}_{\text{cyc}}$	25.8
	w/o $\mathcal{L}_{\text{eq}}$ & $\mathcal{L}_{\text{cyc}}$	26.1

Table 3.4: **Ablating the losses.** We assess the contributions of the losses removing them one at a time. On the last row, we remove $\mathcal{L}_{\text{eq}}$ and $\mathcal{L}_{\text{cyc}}$ to show that although both address symmetry, they are complementary. Ablating $\mathcal{L}_{\text{syn}}$ falls under Table 3.3 (w/o Synth data) as it uses different data. We show the average score over all classes.

class varies from 2,899 for horses (most) to 735 for bears (least). In comparison, competing methods train on more images. [N. Kulkarni et al. 2020] train on combined PASCAL and ImageNet images, and the supervised method of [Neverova et al. 2021] is trained on DensePose-LVIS and DensePose-COCO, the latter of which consists of 5 million image-to-template annotations for humans. Likely due to the density of the pseudo-ground truth, our method only needs a much smaller amount of data and can be trained with as few as a few hundred images (e.g., bear class from DensePose-LVIS, or the models we show in Fig. 3.1). Similarly to CSM [N. Kulkarni et al. 2019; N. Kulkarni et al. 2020] and CSE [Neverova et al. 2021], we use masks for training.

3.4.3 Evaluation

We evaluate our models and CSE [Neverova et al. 2021] on DensePose-LVIS using the geodesic error. We normalize the maximum geodesic distance on each mesh to 228, following [Riza Alp Güler et al. 2018; Neverova et al. 2020], and use a heat solver to obtain all vertex-to-vertex geodesic distances. Additionally, we use PF-PASCAL [Ham et al. 2017] to evaluate our model on keypoint transfer. PF-PASCAL consists of pairs of images with annotated salient keypoints (e.g., left eye, nose, *etc.*), and we evaluate image-to-mesh-to-image correspondences using the image-to-image annotations. We use the test split of [Junyi Zhang et al. 2023a] and evaluate using PCK 0.1, following prior work [Junyi Zhang et al. 2023a;

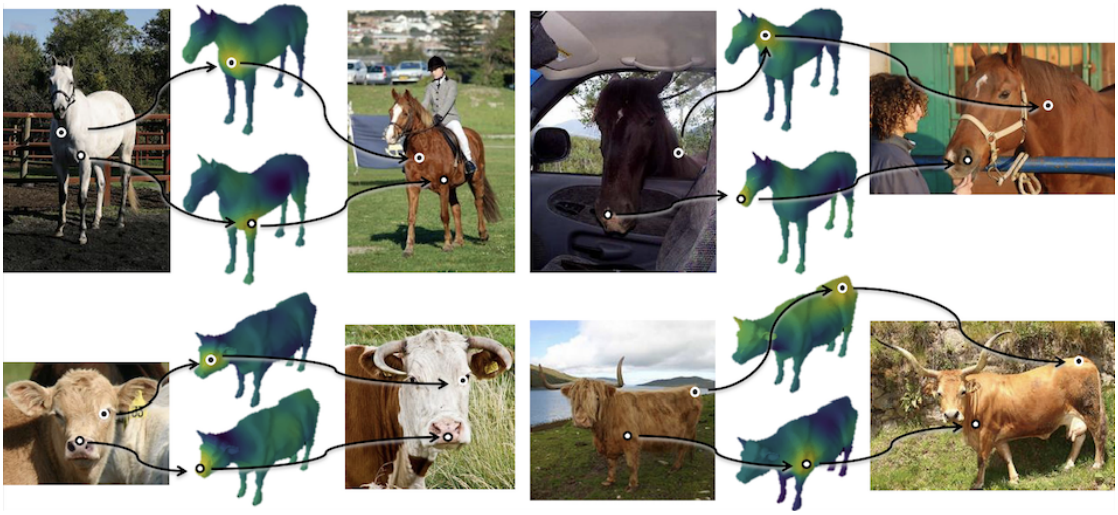


Figure 3.7: **Image-to-image correspondences.** We show image-to-image correspondences on PF-PASCAL, which we find using pixel-to-vertex-to-pixel matching. The heatmaps on the shape show the similarity from the source image location to every vertex.

Neverova et al. 2021; N. Kulkarni et al. 2020].

Image-to-shape correspondences. In Table 3.1 we compare the quality of the image-to-template correspondences established by SHIC, by our zero-shot method based on SD-DINO, and by CSE [Neverova et al. 2021], which is supervised, on the DensePose-LVIS dataset. The most important result is that SHIC learns better canonical maps than CSE despite using no supervision. While the training images of SHIC and CSE are the same and the pseudo-ground truth is noisy (Zero-shot$_{\text{SD-DINO}}$ is what we use for supervision), this result can be explained by the fact that our automated supervision is significantly denser than the manual labels collected by [Neverova et al. 2021] (just three per image).

We show some qualitative results on this dataset in Fig. 3.6, where we color every point on the image according to the corresponding color on the mesh. The regularity of the remapped texture illustrates the quality of the correspondences. Once more, the learned canonical map (ours) is significantly better than the pseudo-ground truth (SD-DINO). Compared to CSE, SHIC performs similarly and tends to have a more regular structure on the heads. We show more qualitative evaluations and failure cases in the Appendix.

Ablation study. We ablate the components of our method in Table 3.3 and Table 3.4. When using mean pooling instead of max pooling for obtaining the

pseudo-ground-truths, we see a significant drop in performance. This is because image-to-image correspondences are more reliable when the objects are in similar poses, and with max pooling we only get contribution from the most similar view. Next, we look at the importance of different sources of supervision. When we remove the pseudo-ground-truth from our synthetic pipeline (Section 3.3.5), the model loses performance. When we exclude all natural images (w/o LVIS), and only train on synthetically generated data, we see a more pronounced drop in performance. Finally, we exclude the pseudo-ground-truth from SD-DINO (Section 3.3.3) and thus only train with the synthetic data and the cycle consistency and equivariance losses on natural images. In this case, the model learns a degenerate solution, where for natural images it only predicts vertices on one side of the shape (i.e. left or right). Such degenerate solutions have been observed by [Truong et al. 2021] for unsupervised image-to-image matching when using cycle consistency losses. This shows the importance of using our pseudo-ground-truth. Finally, we ablate the losses we use in Table 3.4. All losses are necessary for the final performance. We find that $\mathcal{L}_{\text{pseudo}}$, where we frame pixel-to-vertex matching as a multi-class classification problem, has a bigger contribution than the distance-aware loss $\mathcal{L}_{\text{dist}}$ of [Neverova et al. 2021]. Additionally, we see that both losses that address symmetry, $\mathcal{L}_{\text{eq}}$ and $\mathcal{L}_{\text{cyc}}$, improve performance, and are complementary to each other.

Keypoint transfer. Next, we evaluate SHIC on the PF-PASCAL [Zuffi et al. 2017] in Table 3.2. In this dataset, one evaluates the quality of image-to-image correspondences instead of image-to-template. There are two ways of using our method to induce image-to-image correspondences. The first is to use the learned canonical maps, transferring points from one image to the template and then back to the other image. The second is to directly match the image-based CSE embedding learned in Section 3.3.4. The key findings from our results are: (1) SHIC outperforms the supervised versions of Rigid [N. Kulkarni et al. 2019] and Articulated [N. Kulkarni et al. 2020] CSM, as well as CSE [Neverova et al. 2021] and greatly outperforms all unsupervised approaches. (2) The image-to-image correspondences induced by the canonical maps are significantly better than the ones induced by the image-based CSE embeddings, once again illustrating the importance of the canonical maps. We show qualitative examples in Fig. 3.7.

Novel classes. SHIC can be trained on any class as long as there is a collection of a few hundred images and a suitable template mesh. In Fig. 3.1 we show qualitative results from a model we train on two classes — T-Rex and Appa, a six-legged flying bison from Avatar: The Last Airbender, using 480 and 180 manually collected images, respectively. We extract masks for training using the open vocabulary segmentation method of [S. Liu et al. 2023]. Although the 3D template for Appa is toy-like and does not resemble the images closely, and we only use 180 images, SHIC still manages to learn useful correspondences. This is a considerable advantage of our method over supervised previous work, as it can be trained from a small number of images and without human supervision. This allows the construction of general-purpose shape correspondence models for almost any category.

3.5 Conclusion

We have introduced an unsupervised method to learn correspondence matching between a 3D template and images. Critically, this model can be trained without supervision and from less than 200 images, which makes it applicable to a vast number of objects. This is a significant step beyond previous work that required lots of manually labelled correspondences. We hope that SHIC will enable many downstream tasks where learnt robust correspondence estimation was previously impossible.

Ethics. We utilize the DensePose-LVIS dataset [Neverova et al. 2020] and PF-PASCAL [Ham et al. 2016] for evaluation in a manner compatible with their terms. The images may contain humans, but we only consider occurrences of animals and there is no processing of biometric information. For further details on ethics, data protection, and copyright please see `https://www.robots.ox.ac.uk/~vedaldi/research/union/ethics.html`.

Acknowledgements. We thank Orest Kupyn for helpful discussions and Luke Melas-Kyriazi for proofreading this paper. A. Shtedritski is supported by EPSRC EP/S024050/1. A. Vedaldi is supported by ERC-CoG UNION 101001212.

3.6 Appendix

In this Appendix, we first discuss the limitations of our approach (Section 3.6.1). Then we discuss implementation details (Section 3.6.2) and provide additional ablations (Section 3.6.3). Finally, we show more qualitative examples (Section 3.6.4) and show a failure mode we observe (Section 3.6.5).

3.6.1 Limitations

Our method has several limitations. First, it relies on having a few hundred images per category, which might not always be possible for low-resource classes. However, this is still a significant step forward from prior works, which need much more data and/or human annotations.

Next, the symmetry equivariance loss we propose assumes the shape is symmetric. While this is true for all shapes we consider, there could be several instances where this assumption does not hold: (i) if the shape is not symmetric by design, e.g., it an animal that misses a leg; (ii) if the shape is articulated and thus not symmetric. In that instance, the loss $\mathcal{L}_{eq}$ should not be used, which would lead to a small drop in performance.

Finally, our model only predicts image-to-vertex matching, whereas prior methods such as CSE [Neverova et al. 2021] also predict segmentation masks. However, prior methods do *not* evaluate segmentation performance, as they are not competitive, and this is not the main point of the methods. Furthermore, they use masks as an *additional form of supervision*, as the model is additionally trained to predict masks, whereas we only use masks to sample points used during training (as not to try matching background points to the shape).

3.6.2 Additional implementation details

Symmetry equivariance loss

We automatically discover the plane of symmetry of the shape. We assume the shape's plane of symmetry is either one of the (x, y, z) planes. In practice, this is most often true. We test each of the (x, y, z) planes as follows. First, we center the mesh. Then, for every plane, we mirror all vertices along that plane.



Figure 3.8: **Background images.** To generate synthetic images, we sample from these, do a random crop, and predict depth.

For every vertex, we find its nearest neighbour mirrored vertex. We sum the Euclidean distances between all vertices and their mirrored nearest neighbours. Intuitively, the correct plane of symmetry corresponds to the smallest sum of distances. Finally, for every vertex x, we obtain its symmetric one x_F by finding its nearest neighbour when we mirror the shape along the selected plane of symmetry.

Training

During training, we perform data augmentations: random crops, rotations, and colour jitters. We perform these on both the natural and synthetic (generated with a depth-to-image model) images. We train using the Adam optimizer for 40 epochs, using $lr = 0.001$, which is decreased $10\times$ after 20 epochs.

Synthetically generated ground-truth

As discussed in the paper, to generate each synthetic image, we sample a viewpoint and a background.In practice, we sample from 4 background images (Fig. 3.8), which we randomly crop before computing depth.We find that we can obtain diverse backgrounds with a small number of background templates by using different random seeds. We sample viewpoints only from the side and front.We found that when we sample an image from the back, Stable Diffusion still tries to place a face on the back of the head, leading to unnatural-looking images. We show more examples of generated images in Fig. 3.9.



Figure 3.9: **Synthetically generated images.**

3.6.3 Additional ablations

Pseudo-ground truth

We perform additional ablations on the features used to construct the pseudo-ground-truth Σ in Table 3.5. First, we render shaded surfaces instead of surface normals and find that leads to a small drop in performance. Next, we exclude the SD features from SD-DINO [Junyi Zhang et al. 2023a], and only use DINO features for matching.This makes computing the pseudo-ground-truth Σ faster, as SD features are more expensive.As expected, we see decreased performance when only using DINO features.

Number of training images

We train our method using a different number of natural images in Table 3.6. We train on $\{50, 200, 500, \text{and } 2k+\}$ images, where $2k+$ is the number of images of the particular class in the dataset, falling between 2k and 3k. We exclude the classes "bear" and "sheep" as they contain under $2k$ images.We see that with as few as 500 images, we achieve comparable performance to our full models.

Ablation		DensePose-LVIS
	Ours	**24.9**
Renders	`shaded` (instead of `normals`)	25.2
Features	w/o SD	25.4

Table 3.5: **Data ablations.** First, we ablate using shaded renders of the template shape instead of surface normals. Next, we train models using *only* DINO features (w/o SD), as they are quicker to compute. We evaluate using geodesic distance (lower is better).

No# images	DensePose-LVIS
50	34.7
200	29.8
500	24.9
2k+	22.3

Table 3.6: **Ablation of the number of training images.** We ablate the number of training images used for each class. For this ablation, we exclude the "bear" and "sheep" classes as they have under 2k images, the other classes have between 2k and 3k images. We evaluate using geodesic distance (lower is better).

3.6.4 Qualitative examples

In Fig. 3.10 we show similarity heatmaps of the visual feature with the CSE embeddings over the shape. We show further qualitative examples of texture remapping in Figs. 3.11 to 3.13.

3.6.5 Failure case

We observe a failure case, where the model predicts wrong patches (Fig. 3.14). We notice that these patches correspond to the same semantic part, but on opposite sides (e.g., a patch of "left belly" is predicted where there should be "right belly").



Figure 3.10: **Similarity heatmaps.** We show similarity heatmaps between the visual feature sampled at the annotated location in red with the CSE embeddings learnt over the shape. We color every vertex according to that similarity and annotate the most similar vertex in red.

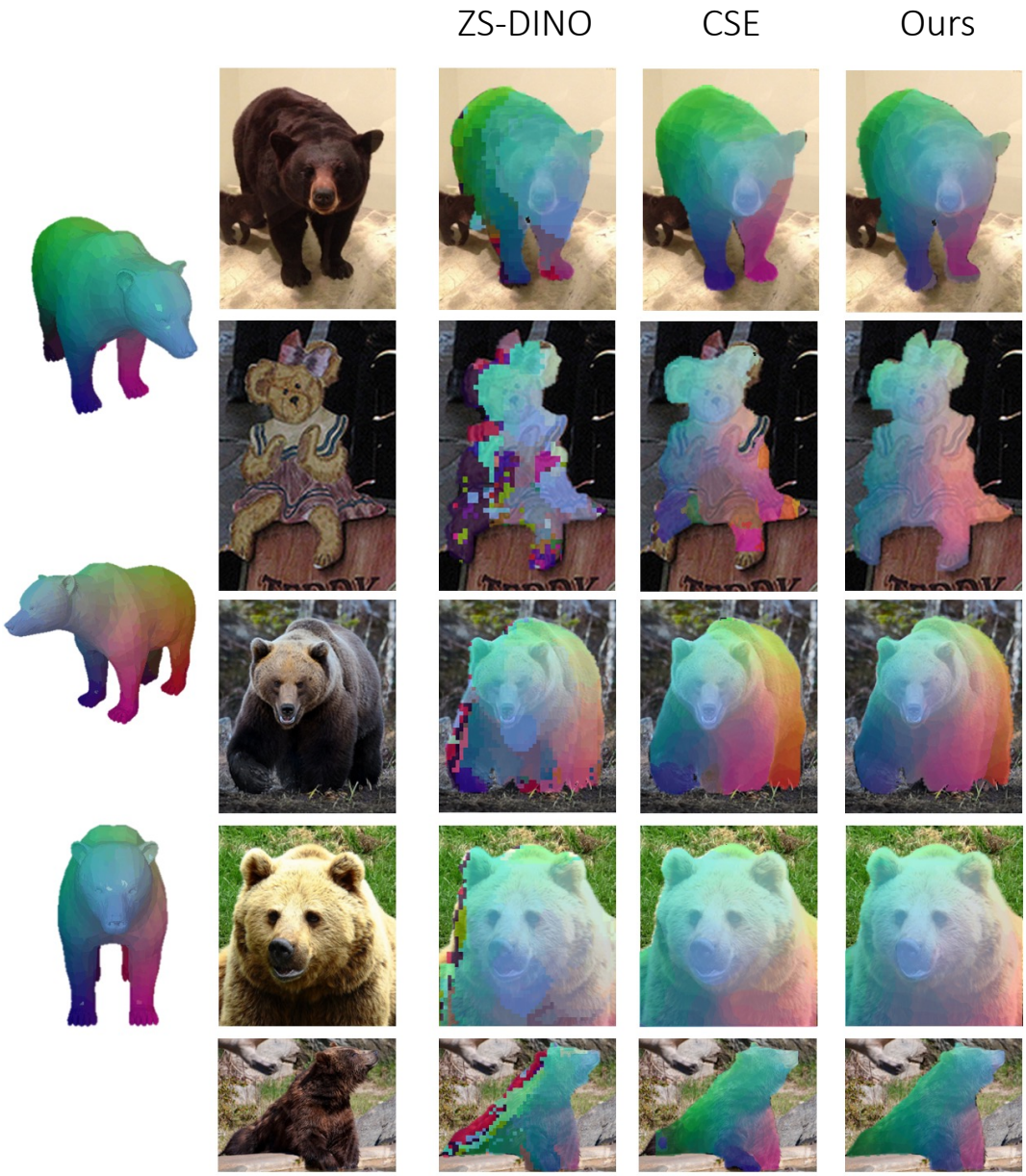


Figure 3.11: **Qualitative results.**

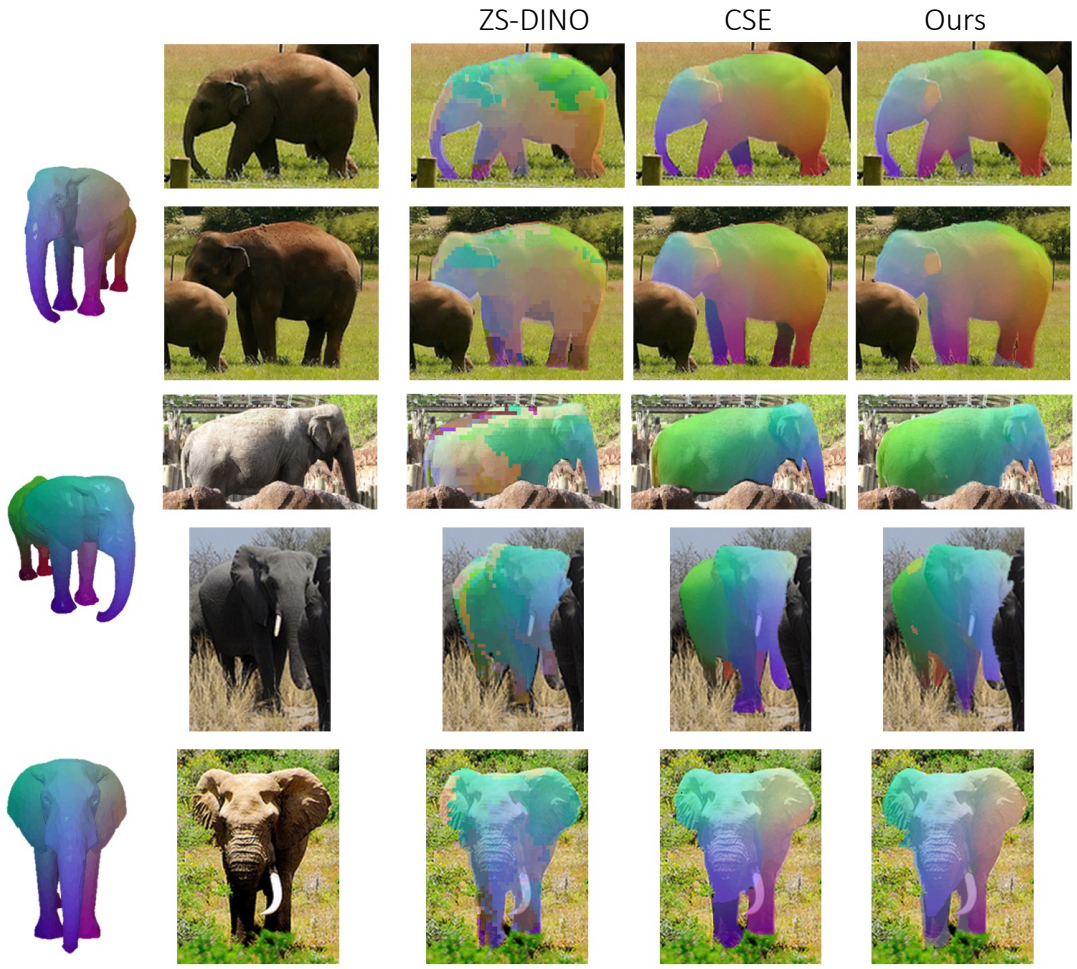


Figure 3.12: **Qualitative results.**



Figure 3.13: **Qualitative results.**

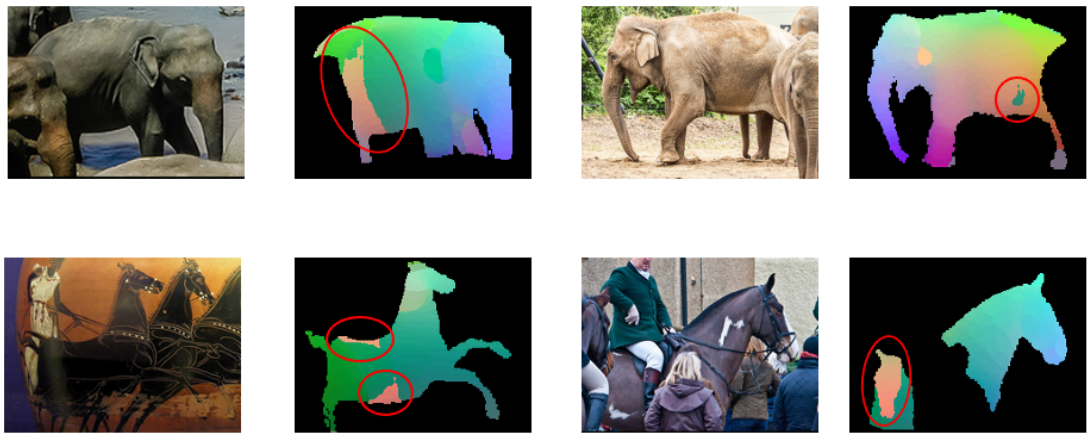


Figure 3.14: **Failure cases.** We annotated failure cases in red, where the model predicts wrong patches.

Chapter 4

What Does CLIP Know About a Red Circle? Visual Prompt Engineering for VLMs

The paper was presented at ICCV 2023 as an oral presentation.

What does CLIP know about a red circle? Visual prompt engineering for VLMs

Aleksandar Shtedritski Christian Rupprecht Andrea Vedaldi

Visual Geometry Group, University of Oxford

{suny, chrisr, vedaldi}@robots.ox.ac.uk

September 29, 2025

Abstract

Large-scale Vision-Language Models, such as CLIP, learn powerful image-text representations that have found numerous applications, from zero-shot classification to text-to-image generation. Despite that, their capabilities for solving novel discriminative tasks via prompting fall behind those of large language models, such as GPT-3. Here we explore the idea of visual prompt engineering for solving computer vision tasks beyond classification by editing in image space instead of text. In particular, we discover an emergent ability of CLIP, where, by simply drawing a red circle around an object, we can direct the model's attention to that region, while also maintaining global information. We show the power of this simple approach by achieving state-of-the-art in zero-shot referring expressions comprehension and strong performance in keypoint localization tasks. Finally, we draw attention to some potential ethical concerns of large language-vision models.

4.1 Introduction

Large Language Models (LLMs) such as GPT-2/3 [T. Brown et al. 2020; Radford et al. 2019] and ChatGPT [*ChatGPT* n.d.] have demonstrated surprising emerging behaviours. For example, these models can perform language translation without being explicitly trained for it, in a zero-shot manner. This can be partially explained by the fact that occurrences of the desired behaviours, such as translating

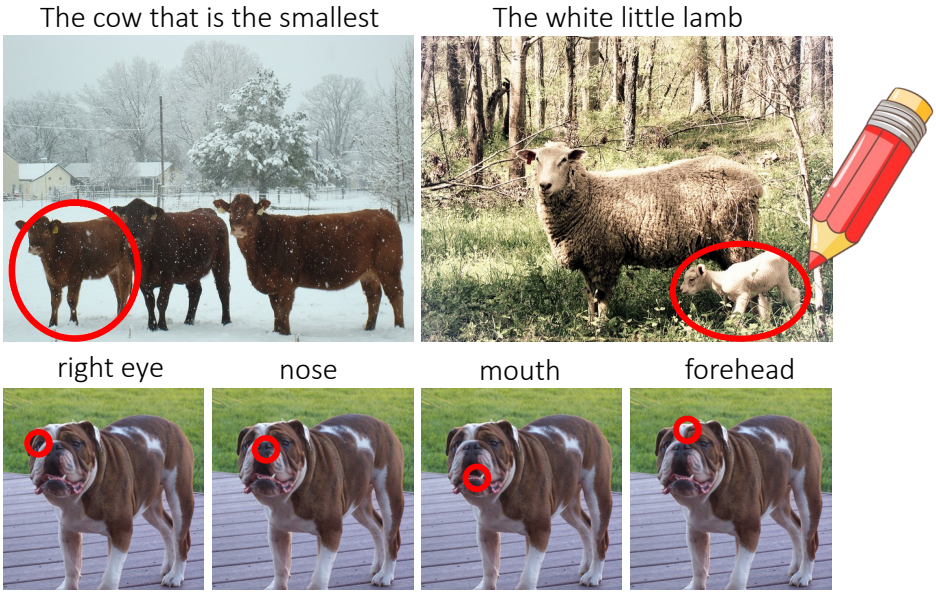


Figure 4.1: **Visual Prompt Engineering.** We draw multiple annotations over an image and have CLIP choose the correct one given a caption. Here we show predictions for the given expressions. Top: Examples from RefCOCOg on referring expressions detection. Bottom: Example from SPair71k on keypoint localization.

between two languages, naturally occur in their enormous training corpus, which is, essentially, the Internet.

Interesting emergent behaviours have been observed in large Vision-Language Models (VLMs) like CLIP [Radford et al. 2021] too. For example, CLIP can be used for zero-shot classification by checking the compatibility of a given image with prompts such as "an image of a X", where X is one of a set of class hypotheses to be tested.

Emergent behaviours are elicited by supplying suitably crafted inputs to the VLMs, often called *prompts*. As in the example above, researchers have mostly focused on *engineering textual prompts*, manipulating the textual input of the model. This approach is inspired by LLMs, where manipulating the textual modality is the only available option. However, VLMs are inherently multimodal and offer the possibility of manipulating both modalities, textual and visual. While the textual modality is the natural choice for expressing semantics, the visual modality can be better for expressing geometric properties such as location.

In this paper, we thus explore *visual prompt engineering* [1]. We do so with two

[1]Note the difference between visual prompt *tuning*, a setting previously explored, where the prompts are task-specific learnable tokens, and visual prompt *engineering*, where we apply a fixed augmentation in pixel space.

goals. The first goal is to contribute one more practical tool for extracting useful information from VLMs in a zero-shot manner. We demonstrate this by obtaining state-of-the-art zero-shot results in referring expressions comprehension by engineering visual prompts. The second goal is to characterise interesting and unexpected properties of the VLMs and their training data, including identifying some behaviours that can raise ethical concerns.

Perhaps the most surprising of our findings is the effectiveness of a particular type of visual prompting: **drawing a plain red circle** on top of the image (Fig. 4.1). We show that this simple intervention steers the VLM to analyse/talk about the image region contained in the circle. This behaviour can then be used for tasks such as naming a specific object or object part or detecting particular image regions based on a description. The latter, for instance, is achieved by marking each object proposal with a red circle and using the VLM to find the best match with respect to the provided referring expression, achieving strong results on multiple benchmarks in the unsupervised regime. Furthermore, we show that prompting with a circle also works for finer-grained localization, marking specific object parts or keypoints instead of just whole objects.

We further contrast marking an image with the alternative of cropping it, which, from sliding window classifiers to region neural networks, is the canonical approach to steer the focus of an image-level predictor to a particular image region. We show that, for VLMs at least, marking is significantly more effective than cropping, possibly because it does not lose contextual information like the latter.

Apart from the practical applications, our findings reveal unexpected and intriguing properties of VLMs. We show empirically, that marking with a red circle is optimal among a selection of possible markers (variants of the circle, boxes, arrows, etc.). Presumably, the VLMs understand red circles out of the box because these appear sufficiently frequently in the training corpus, i.e., the Internet. While we do not have access to the full training data of CLIP, we corroborate this intuition by seeking examples of such images in YFCC15M, a dataset of CC-BY images.

Our analysis shows that red circles are indeed present even in a (comparatively small) dataset of images like YFCC15M, but they are *rare*. It is a testament to the extraordinary capacity of VLMs that such a behaviour can be learned from

such rare events, without an explicit focus on doing so. We test models of different sizes/capacities and show that only the larger models exhibit this behaviour reliably, which well corresponds to our intuition.

Finally, we note that the ability of VLMs to learn even from rare events such "red circles" can acquire both desirable and undesirable behaviours. Red circles, in particular, can have a negative connotation in the training data as they are often used by news outlets to mark missing people or criminals and, evidently, the model learns from such examples. As a result, we show that drawing a red circle in an image increases the probability that the model would characterise a person as a criminal or as a missing person.

To summarise, we make the following main contributions: (1) We propose marking as a new form of visual prompt engineering which is effective in extracting useful emergent behaviours in VLMs like CLIP; (2) We use the latter to achieve state-of-the-art zero-shot referring expressions comprehension using a VLM; (3) We provide an analysis of why marking is effective for these models, and link that to the training data and large model capacity; (4) We show that visual prompt engineering can also elicit unwanted behaviours, such as triggering problematic biases in the VLMs, revealing potential ethical issues.

4.2 Related work

Emergent Behaviour from Large Scale Pretraining has mainly been observed in Large Language Models (LLMs). Most notably, GPT-2 [Radford et al. 2019], GPT-3 [T. Brown et al. 2020], and ChatGPT [*ChatGPT* n.d.] have been shown to be capable of tasks such as zero-shot translation, question answering, arithmetic, as well as planning actions for embodied agents [W. Huang et al. 2022]. Fine-tuning LLMs can also lead to models that can generate code from docstrings [Mark Chen et al. 2021] or solve math problems [Cobbe et al. 2021; Lewkowycz et al. 2022]. Only a few emergent zero-shot behaviours have been reported for VLMs like CLIP, mainly for classification [Radford et al. 2021] and OCR [Materzyńska et al. 2022]. Generative VLMs like FLAMINGO [Alayrac et al. 2022] and BLIP [Junnan Li et al. 2023] excel in captioning and visual question-answering tasks, but also have no way of solving pixel-level computer vision tasks.

Prompting VLMs is most commonly performed by prepending a set of learnable tokens to the text input [Z. Guo et al. 2022; Ju et al. 2022; K. Zhou et al. 2022a; K. Zhou et al. 2022b], vision input [Jia et al. 2022; Tu et al. 2022; S. Zhang et al. 2022], or both text and vision inputs [Shen et al. 2022; Zang et al. 2022], in order to easily steer a frozen CLIP model to solve a desired task. [Bahng et al. 2022] learn augmentations in pixel space, such as padding around the image, or changing a patch of the image, which are optimized with gradient descent on a downstream task. [Bar et al. 2022] cast image inpainting as a visual prompting task, using a generative model trained on figures from academic papers. Colorful Prompt Tuning (CPT) [Y. Yao et al. 2021] color regions of an image and use a captioning model to predict which object in an image an expression refers to by predicting its color. Similarly to CPT, we augment the input image in pixel space and perform zero-shot inference. However, we *annotate* the image in a human-like manner and show that our method is more powerful and more flexible than CPT.

Referring Expression Comprehension (REC) aims to localize a target object in an image that corresponds to a textual description. Most approaches to REC start with object proposals, for example, generated with Faster-RCNN [Shaoqing Ren et al. 2015], and learn to score them [Hu et al. 2016; R. Luo and Shakhnarovich 2017; Yongfei Liu et al. 2020a; Sibei Yang et al. 2019a; Peng Wang et al. 2019a]. REC is sometimes considered together with referring expression generation — the task of generating a description of a given region. [R. Luo and Shakhnarovich 2017] use a comprehension model to guide a generator, whereas [Y.-W. Chen et al. 2019] jointly train a detector with a caption generator. Some works model the scene as a graph [Yongfei Liu et al. 2020a; Sibei Yang et al. 2019a; Peng Wang et al. 2019a] or use language parsers and grammar-based methods [Cirik et al. 2018; Daqing Liu et al. 2019], leading to a more interpretable result. More recently, transformer architectures have been used [Muchen Li and Sigal 2021; Kamath et al. 2021; Dou et al. 2022; Zhengyuan Yang et al. 2021; Zhaoqing Wang et al. 2022]. [Muchen Li and Sigal 2021; Kamath et al. 2021; Dou et al. 2022; Zhengyuan Yang et al. 2021] perform text-modulated object detection, where a transformer decoder takes the referring expression as an input and predicts a bounding box. [Zhaoqing Wang et al. 2022] train with a text-to-pixel contrastive loss, which allows for a text-driven segmentation or detection at test time.

Unsupervised Referring Expression Comprehension is a less explored area, only made possible with the introduction of large pretrained vision-language models such as CLIP [Radford et al. 2021]. ReCLIP [Subramanian et al. 2022] crops object proposals and ranks them using CLIP before an ad-hoc postprocessing step to take into account relations such as left/right, smaller/bigger, etc. CPT [Y. Yao et al. 2021] colors object proposal boxes and use a pre-trained captioning model [P. Zhang et al. 2021] to auto-regressively predict which colored proposal corresponds to the query description. Pseudo-Q [H. Jiang et al. 2022] generates descriptions for multiple objects in an image, which is used to train a REC network. However, this model is not fully unsupervised as the pseudo descriptions it uses are generated using a captioning model trained on COCO.

Visual Reasoning Using Large Pretrained Models has been an area of significant interest in the last few years. In addition to referring expression detection [Subramanian et al. 2022], CLIP [Radford et al. 2021] has been used for semantic segmentation [Pakhomov et al. 2021; F. Liang et al. 2022]. [Pakhomov et al. 2021] use CLIP to assign text labels to object parts after doing part co-segmentation in the latent space of a GAN. [F. Liang et al. 2022] utilize CLIP for open-vocabulary segmentation by using a general-purpose mask proposal network and CLIP as a classifier. CLIP has also been used for unsupervised object proposal generation [H. Shi et al. 2022] and open-set detection [Esmaeilpour et al. 2021]. Semantic segmentation also emerges from image only [Melas-Kyriazi et al. 2022b; Y. Wang et al. 2022b] or image-text [Jiarui Xu et al. 2022] self-supervision.

Bias of VLMs is an increasingly popular area of research, as downstream applications come with the risk of perpetuating biases and stereotypes existing in the training data. However, methods for assessing the bias of a VLM are still not well established. [Agarwal et al. 2021] measure the misclassification rate of CLIP of faces of people of different races with non-human and criminal categories, whereas [H. Berg et al. 2022; Chuang et al. 2023; Jialu Wang et al. 2021] measure fairness in retrieval results. Here, we show a different kind of bias, where the addition of a red circle over a person can trigger a negative connotation.

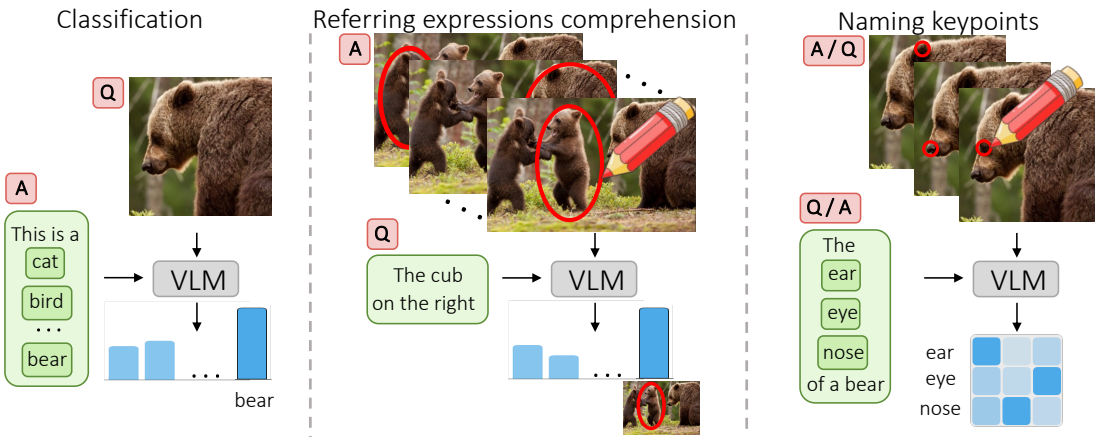


Figure 4.2: **Prompt engineering for VLMs.** We cast zero-shot inference with VLMs as a Q/A problem, each requiring specific prompt engineering. In the figure, Q is *Question* and A is *Answer* (a set of possible answers). Left: text prompt engineering for classification. This widely used method can be interpreted as follows in our framework: The image is the question, and classes are the available answers, which are engineered into prompts. Middle: visual prompt engineering for referring expressions comprehension. The question is the referring expression, and the available answers are the box proposals, which we engineer into visual prompts. Right: visual and text prompt engineering for keypoint matching. For keypoint localization, we use a similar setup to referring expressions, where the question is a keypoint in plain text and the possible answers are all 2D locations in the image.

4.3 Method

Our goal is to develop visual prompting in *Vision-Language Models* (VLMs). VLMs solve prediction tasks that involve jointly processing text and images. For example, models such as CLIP are trained to *match* text and image samples. The input to such a VLM is an image $i \in \mathbb{R}^{3\times H\times W}$ and text $t \in \Sigma^*$, where Σ is an alphabet. The output is a score $s(i,t)$ that expresses the degree of compatibility between the supplied image and text.

4.3.1 Prompt engineering

One of the most striking capabilities of VLMs is their ability to solve a variety of classification tasks with little to no further training at all, in a zero-shot manner. This is done by reducing the task of interest to that of evaluating the VLM on suitably-engineered image and text pairs.

For example, given an image-caption pair (i,t), consider the problem of localizing a named object keypoint in the image. We can cast this as a question-answer problem, where the question $q \in Q$ is the name of the object keypoint (e.g., "right

ear", "front left leg", ...) and the answer $a \in A$ is one of a discrete set of image locations.

Because the VLM computes a compatibility score $s(i, t)$ between an image i and the text t, it cannot be used to map the question q to the answer a directly. However, via prompt engineering, we can use the VLM to construct a compatibility score $s(q, a|i, t)$ between question and answer, conditioned on the input image-text pair (i, t). This score is in general given by the expression

$$s(q, a|i, t) = s(i_{qa}, t_{qa}) \tag{4.1}$$

where i_{qa} and t_{qa} are versions of the input image and text, obtained by transforming the latter to reflect the question-answer pair (q, a).

The specific way Eq. (4.1) should be applied to a problem depends on the specific nature of the latter. For example, in the problem of localizing the named keypoints, it is natural to encode the name of the keypoint via the textual modality and its 2D location via the visual modality. For instance, in order to answer the question $q =$ "right ear" for a given input image i with caption $t =$ "dog", we can engineer the textual prompt $t_{qa} = t_q =$ "an image of the right ear of a dog" to encode a description of the named entity. Likewise, we can engineer the visual prompt $i_{qa} = i_a$ in such a way as to 'select' the location a in the image, using one of the methods discussed in Section 4.3.2. With this, we can answer the question by finding $\hat{a}(q|i, t) = \text{argmax}_{a \in A} s(q, a|i, t)$, that maximizes the score $s(q, a|i, t) = s(i_a, t_q)$, which specializes Eq. (4.1).

In the following sections, we provide further details and apply these ideas to a few concrete tasks.

4.3.2 Visual prompting via marking

The usual way of encoding location information in a visual prompt is to crop the image around the desired location, meaning that i_a is the image cropped around a. This idea has been used extensively with VLMs, including to interpret referring expressions, where maximizing a score of the form $s(i_a, t_q)$ seeks for the image crop that best matches the referring expression t_q.

In this paper, we explore an alternative approach for visual prompting that uses the concept of *marking* the desired region in the image. Marking quite literally means overlaying to the image i_a a circle, a box, or an arrow, which visually indicates the desired location a.

While the idea of marking may sound strange, it is interesting for two reasons. First, differently from cropping, a marked image i_a preserves almost all the information contained in the input image i, including contextual information that crops lack. Second, we show that marking works well with VLMs, *outperforming* cropping-based prompt engineering in some prediction tasks.

While the simplest marking consisting of a red circle is particularly effective, in Section 6.4 we explore several different ways of generating markings. We refer the reader to that section for further details and examples.

4.3.3 Tasks

We study the idea of mark-based prompt engineering by considering several zero-shot prediction tasks, from simple tasks such as matching keypoints to their names to more complex ones such as referring expression comprehension.

Naming Keypoints. The first and simplest task that we consider is matching the name of the keypoints of an object to their 2D locations in an image. The input is an image i, a set of keypoint names Q, and a set of corresponding keypoint locations $A \subset \{0, \dots, H-1\} \times \{0, \dots, W-1\}$. The number of names and locations is the same ($m = |Q| = |A|$) and the goal is to match the two. We express the latter as predicting the square permutation matrix $\Pi \in S_m$ that associates each name q to its corresponding location a (i.e., $\Pi_{qa} = 1$).

In order to predict Π, we use Eq. (4.1) to define the cost of associating name q to location a as $C_{qa} = s(i_a, t_q)$ where i_a is obtained either via cropping or marking and t_q is just the name of the keypoints prefixed by the string "an image of". For this problem, the role of questions and answers is symmetric and we decode the cost matrix C into a permutation matrix Π via optimal transport:

$$\hat{\Pi}(i, Q, A) = \operatorname*{argmax}_{\Pi \in S_m} \sum_{q \in Q, a \in A} \Pi_{qa} \exp\left(-\tau C_{qa}\right)$$

where $\tau > 0$ is a temperature parameter. This optimization problem is solved efficiently via the Sinkhorn-Knopp algorithm [Sinkhorn and Knopp 1967], which renormalizes matrix C.

Keypoint Localization. The second task is a more useful and difficult variant of the first. The goal is still to localize a named keypoint q in an image, but this time the locations A are a subset of a $m \times m$ regular grid. These are further restricted to a salient image region extracted by using the unsupervised saliency method of [Y. Wang et al. 2022b] to avoid testing irrelevant locations in the background. The difference compared to naming keypoints is that this version of the problem does not assume prior knowledge of the possible locations of the keypoints. Given the name q of a keypoint, its location a is then obtained as $\hat{a}(i, q) = \text{argmax}_{a \in A}\, s(i_a, t_q)$ where i_a and t_q are as defined previously.

Method	Name-to-keypoint							Keypoint-to-name						
	CUB	**Spair71k**						**CUB**	**SPair71k**					
Random	8.2	16.8	15.0	10.5	9.4	15.1	11.9	8.2	16.8	15.0	10.5	9.4	15.1	11.9
Crop w/o SK	15.8	28.5	28.5	28.5	20.1	26.1	29.7	18.7	22.4	19.1	24.0	14.9	27.3	25.1
Crop w/ SK	25.5	35.1	37.5	34.6	23.9	32.9	36.3	25.8	36.1	32.5	32.7	19.8	35.3	32.5
Red Circle w/o SK	46.5	54.8	53.1	51.6	40.1	47.4	45.2	29.5	26.8	24.9	36.9	18.8	31.8	28.9
Red Circle w/ SK	**58.2**	**67.6**	**60.1**	**59.3**	**53.1**	**56.7**	**52.8**	**56.5**	**67.2**	**54.4**	**59.7**	**49.8**	**56.6**	**53.0**

Table 4.1: **Naming keypoints results on CUB and SPair71k**. On SPair71k, we show results on all animal classes — bird, cat, dog, horse, sheep, cow. We show the percentage of correctly matched keypoints and names, given a list of them. We compare to randomly guessing the correct correspondence and cropping around the region of interest, rather than drawing an annotation. We compare all methods with and without normalization with the Sinkhorn-Knopp (SK) algorithm.

Referring Expression Comprehension. Comprehending a referring expression means detecting an object in an image that corresponds to a textual specification that explicitly refers to it (e.g., "fourth dog from the right"). Similarly to prior work [Y. Yao et al. 2021; Subramanian et al. 2022; H. Jiang et al. 2022], given an image i, we approach this problem by extracting first a set of object proposals using the method from [L. Yu et al. 2018] and interpret those as the set of possible answers A. The set of questions Q is instead a collection of referring expressions extracted from a given benchmark dataset. For each referring expression, the best matching proposal is then given by

$$\hat{a}(i, q) = \underset{a \in A}{\text{argmax}} \left[s(i_a, t_q) - \frac{1}{|Q|} \sum_{\bar{q} \in Q} s(i_a, t_{\bar{q}}) \right].$$

The engineered prompts i_a and t_q are defined as in Section 4.3.3. In this case, we found it useful to subtract from the score the average with respect to all possible referring expressions Q. This weighs down hypotheses a such as faces that are visually very salient and tend to respond very strongly to all questions q.

4.4 Experiments

We study the properties of visual marking in VLMs by considering first the three tasks of Section 4.3.3: naming keypoints, localizing keypoints, and referring expression comprehension.

4.4.1 Naming Keypoints

Naming keypoints is a comparatively simple problem that has no direct application; however, it is simpler and faster to evaluate than the other tasks, so we use it to ablate various aspects of our method.

Data and implementation details. For this task, we consider the CUB-200-2011 (CUB) [Welinder et al. 2010] and SPair71k [J. Min et al. 2019] datasets. The first contains named keypoint annotations for each image, whereas the second only annotates matching keypoints in pairs of images, but does not name them. We thus augment the latter, manually naming each keypoint instance in each animal image. We further crop the images from SPair71k with the provided bounding boxes. For the VLM, we use the ViT-L/14@336px backbone. Please see the sup. matt. for details.

Results. Recall that, in this task, the output of the predictor is a permutation matrix Π associating each keypoint location to a corresponding name. We report (i) the ratio of keypoint names that are mapped to the correct location and (ii) the ratio of keypoint locations that are mapped to the correct names. To the best of our knowledge, there are no prior works that associate keypoints with their names. We thus compare the result of this new task to (a) random choice and (b) a baseline where i_a is obtained by cropping.

As seen in Table 4.1, prompting via visual marking (red circles) significantly outperforms the baselines, achieving almost twice the accuracy. Using the Sinkhorn-

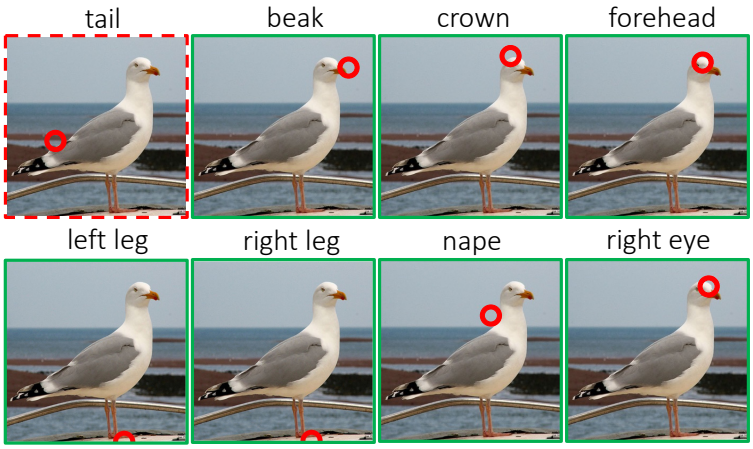


Figure 4.3: **Qualitative Results on Localizing Keypoints** on an image from SPair71k. Green and red (dashed) borders are for correct and wrong predictions according to PCK with $\alpha = 0.1$. The red circle shown is thicker than the one used; see the sup. matt. for examples of the actual thickness.

Marker shape	Mean	Best
Circle	**33.5 ± 4.5**	**46.5**
Arrow	28.3 ± 3.1	36.3
Square	24.1 ± 3.6	36.3
Cross	21.5 ± 6.3	34.5

Circle color	Mean	Best
Red	**36.4 ± 5.1**	**46.5**
Green	34.3 ± 4.2	43.3
Purple	34.0 ± 3.7	41.9
Blue	32.7 ± 3.9	41.1
Yellow	32.4 ± 4.0	40.8

Table 4.2: **Ablation of annotation types for naming keypoint.** We evaluate on CUB and present results across a variety of colors and sizes (for marker shape) and sizes (for circle color). For full results refer to the sup. matt.

Knopp (SK) algorithm to normalize the matching score further boosts results, mainly improving results for points that are ambiguous and close to each other, e.g., mouth and nose.

What is the best visual marker? We compare the use of (i) different shapes for highlighting a location: circle, rectangle, cross, arrow, (ii) different sizes, and (iii) different colors of the annotations, and show some examples in Fig. 4.4. We compare different shapes and colors in Table 4.2 and find that red circles perform best. Red is the best color despite the fact that it is a commonly occurring color in images, unlike colors like purple which can be found less often in nature and can thus be more distinctive, but lead to worse performance. We attribute this to the fact that this emergent capability of CLIP exists due to human-centric manipulations of its training data, and humans are likely to annotate using red circles, as shown next.



Figure 4.4: **Example Annotations for Keypoints.** We show how the annotations we use look for different shapes and colors. We also experiment with multiple thicknesses and sizes. The red circle annotation on the left is the one we use throughout all evaluations unless stated otherwise.



Figure 4.5: **Discovered annotations in YFCC15M.** We show images from YFCC15M that have been human-annotated, and their corresponding captions. Discovered by training a simple detector.

Are there visual markers in the training data? To explore the hypothesis that CLIP can zero-shot classify annotations on images because of similar examples seen during training, we find images in YFCC15M that contain markers (YFCC15M is a subset of the CLIP training data). To this end, we train a binary classifier using an ensemble of a ViT-B/16 and RN50x16 CLIP vision encoders to classify images in YFCC15M that contain annotations. We then use this to filter a 6M subset of YFCC15M and take the top 10k images with the highest score. Finally, we manually examine the 10k images and find 70 images that have annotations drawn on top of them. We show 3 such images in Fig. 4.5. Hence, the training data contains examples of markers, but they are very rare ($\sim$0.001%), suggesting that such behavior can only be learned from very large datasets by high-capacity models. This is further explored next.

How do different VLMs differ? We compare a number of CLIP models in Fig. 4.6. In general, we observe that the performance of keypoint matching improves with (i) the size of the pretraining dataset and (ii) the size of the vision encoder. The former holds true for CLIP models trained on WIT-400M vs YFCC-15M (which is a subset of WIT-400M). However, using the LAION-2B dataset for

Method	Backbone	Data	Params	Name-to-keypoint		Keypoint-to-name	
				CUB	**Spair71k**	**CUB**	**SPair71k**
CLIP [Radford et al. 2021]	ViT-B/32	400M	87M	19.1	26.7	19.1	25.6
CLIP [Radford et al. 2021]	ViT-B/16	400M	86M	22.2	34.0	22.1	33.6
CLIP [Radford et al. 2021]	RN50x16	400M	167M	30.6	41.0	29.7	40.0
CLIP [Radford et al. 2021]	ViT-L/14	400M	304M	47.9	54.3	48.0	51.2
CLIP [Radford et al. 2021]	ViT-L/14⋆	400M	304M	**58.2**	**58.3**	**56.5**	**56.8**
OpenCLIP [Cherti et al. 2022]	ViT-B/32	2B	87M	19.4	27.5	20.7	27.2
OpenCLIP [Cherti et al. 2022]	ViT-L/14	2B	304M	33.9	42.4	33.3	41.5
OpenCLIP [Cherti et al. 2022]	ViT-H/14	2B	632M	45.0	53.7	42.8	50.5
OpenCLIP [Cherti et al. 2022]	ViT-g/14	2B	1.01B	44.2	47.2	42.5	43.6
OpenCLIP [Cherti et al. 2022]	ViT-G/14	2B	1.84B	50.4	52.5	48.9	48.6
SLIP [Mu et al. 2022]	ViT-S/16	15M	22M	13.0	17.8	12.0	16.7
SLIP [Mu et al. 2022]	ViT-B/16	15M	86M	17.3	16.5	16.7	17.1
SLIP [Mu et al. 2022]	ViT-L/16	15M	303M	24.6	26.3	24.1	25.0
CLIP† [Mu et al. 2022; Radford et al. 2021]	ViT-S/16	15M	22M	11.4	16.5	12.4	15.8
CLIP† [Mu et al. 2022; Radford et al. 2021]	ViT-B/16	15M	86M	14.0	18.4	14.8	17.8
CLIP† [Mu et al. 2022; Radford et al. 2021]	ViT-L/16	15M	303M	15.0	20.5	15.8	20.4
FILIP‡ [Yangguang Li et al. 2021; L. Yao et al. 2021]	ViT-B/32	15M	90M	8.9	15.6	8.7	15.6
DeFILIP [Yangguang Li et al. 2021]	ViT-B/32	15M	90M	12.5	19.5	12.5	19.4
CLIP‡ [Yangguang Li et al. 2021; Radford et al. 2021]	ViT-B/32	15M	90M	10.6	14.1	11.0	14.3
DeCLIP [Yangguang Li et al. 2021]	ViT-B/32	15M	90M	15.8	19.9	15.8	18.7
DeCLIP [Yangguang Li et al. 2021]	ViT-B/32	88M	90M	19.4	23.6	19.7	22.3

Table 4.3: **Comparison of backbones and pretraining methods**. We compare several VL pretraining methods, implementations, and backbones. † and ‡ : we evaluate the models trained by [Mu et al. 2022] and [Yangguang Li et al. 2021], respectively. ⋆ : we evaluate a model trained on images of input size 336×336. For SPair71k, we present the average across all animal classes. The 400M, 2B, 15M, and 88M datasets are WIT-400M, LAION-2B, YFCC-15M and DeCLIP-88M, respectively.

pretraining leads to worse results. We suspect this result comes from differences in filtering when creating WIT-400M and LAION-2B, where in the latter, examples of annotations might have been discarded due to a stronger focus on *aesthetic* images for generative models. Similarly, we see big gains in performance as we increase the size of the vision encoder, and the gains do not seem to converge with the biggest available models. We emphasize on the dramatic increase in performance of the WIT-400M pretrained CLIP — the biggest models improve on the performance of the smallest by 250% on ZS keypoint matching, whereas the improvement on ZS ImageNet-1K classification is just 20%. We draw similarities between this task and tasks in the domain of NLP, such as zero-shot or one-shot arithmetic, where only the largest GPT models perform well [T. Brown et al. 2020]. We argue that in a similar fashion, the vision encoder needs sufficient capacity and data in order to show this emergent behavior.

4.4.2 Localizing Keypoints

For this experiment, we use the same data and network architecture as for the previous one, but report the *percentage of correct keypoints* (PCK) as a metric, as

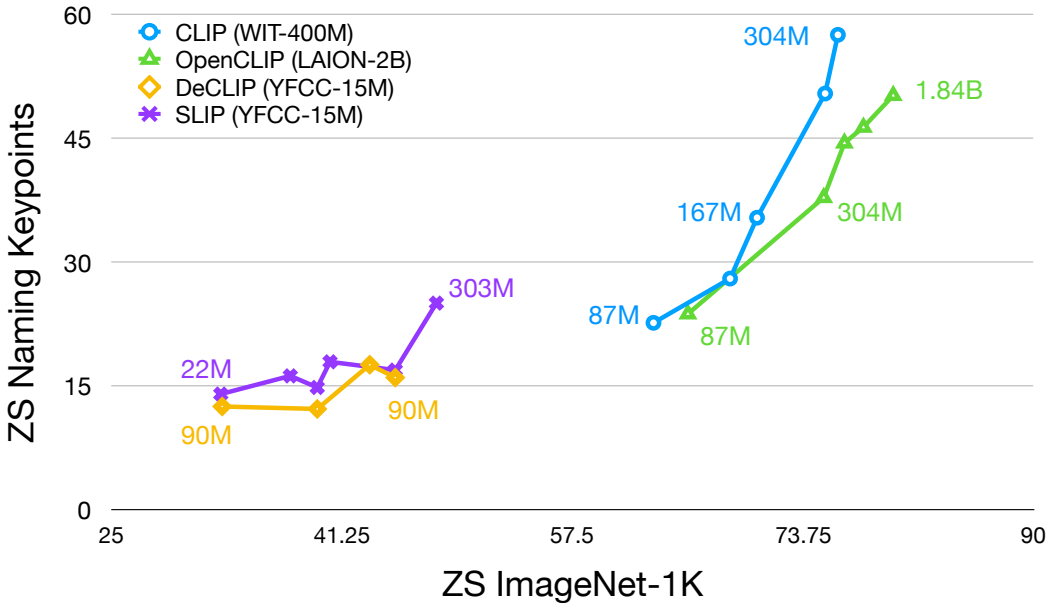


Figure 4.6: **Comparison of Zero-shot (ZS) Naming Keypoints to ZS ImageNet-1K.** We show results for CLIP [Radford et al. 2021], OpenCLIP [Cherti et al. 2022], DeCLIP [Yangguang Li et al. 2021] and SLIP [Mu et al. 2022]. We compute the average text-to-image and image-to-text matching score on SPair71k and CUB and compare it against the reported zeros-shot ImageNet-1K accuracy for each model. For all methods, we show the pretraining dataset and the number of parameters of the vision encoders of some of the models.

Method	**CUB**	**Spair71k**					
Random	8.3	6.7	4.7	4.6	5.5	3.3	4.3
Crop	21.3	28.4	28.8	21.9	16.0	23.4	24.5
Red Circle	**45.2**	**53.4**	**54.5**	**56.4**	**40.9**	**43.2**	**42.2**

Table 4.4: **Named Keypoint Localization Results.** We report PCK of predicted keypoint location and compare to a baseline where we crop around the region.

the latter is widely used when evaluating semantic correspondences. Given a set of ground-truth points $\mathcal{P} = \{p_m\}_{m=1}^{M}$ and predictions $\hat{\mathcal{P}} = \{p_m\}_{m=1}^{M}$, PCK is given by:

$$\text{PCK}(\mathcal{P}, \hat{\mathcal{P}}) = \frac{1}{M} \sum_{m=1}^{M} \mathbb{1}[\|\hat{p}_m - p_m\| \leq \delta].$$

Here, δ is a distance threshold given by $\delta = \alpha \max(H, W)$, where $0 < \alpha < 1$ is a ratio and (H, W) is the bounding box size. For all datasets we use $\alpha = 0.1$. Keypoint localization also utilizes an unsupervised saliency mask to ignore background locations.

Similarly to the naming task, we compare keypoint localization to random guessing

Method	ZS	RefCOCO			RefCOCO+			RefCOCOg	
		Val	TestA	TestB	Val	TestA	TestB	Val	Test
DTWREG	✗	39.2	41.1	37.7	39.2	40.1	38.1	–	–
Pseudo-Q	✗	56.0	58.3	54.1	38.9	45.1	32.1	46.3	47.4
CPT	✓	32.2	36.1	30.3	31.9	35.2	28.8	36.7	36.5
$\text{ReCLIP}_{\text{crop}}$	✓	42.0	43.5	39.0	47.4	50.1	43.9	57.8	57.2
$\text{ReCLIP}_{\text{spatial}}$	✓	45.8	46.1	**47.1**	47.9	50.1	45.1	59.3	**59.0**
Red Circle	✓	**49.8**	**58.6**	39.9	**55.3**	**63.9**	**45.4**	**59.4**	58.9

Table 4.5: **Comparison with state-of-the-art on REC.** We report top-1 accuracy (%). † is the crop-based method of [Subramanian et al. 2022] and ‡ is their method that adds relational resolving. *"ZS"* refers to zero-shot approaches. Drawing red circles outperforms other zero-shot approaches on most benchmarks, including ReCLIP‡ that post-processes results using manually designed relational rules. On RefCOCO+ and RefCOCOg, a red circle also outperforms Pseudo-Q and DTWREG that are not zero-shot and use weak supervision.

and the crop-based baseline. As shown in Table 4.4, using red circles significantly outperforms both; as expected, results are further improved by using saliency to further filter keypoint locations. We show qualitative results in Fig. 4.3.

4.4.3 Referring Expression Comprehension

Datasets and implementation details. Referring expression comprehension is commonly evaluated on the RefCOCO [L. Yu et al. 2016], RefCOCO+ [L. Yu et al. 2016], and RefCOCOg [J. Mao et al. 2016] datasets, all of which consist of images from the MS-COCO dataset [T.-Y. Lin et al. 2014] together with expressions that refer to a unique object in the image, which are also annotated with a bounding box. RefCOCO+ only contains appearance-based expressions, whereas RefCOCO and RefCOCOg contain relation-based expressions (e.g., containing the words left/closer/bigger). The test sets of RefCOCO and RefCOCO+ are split in two, where "testA" and "testB" contain only people and non-people, respectively. We evaluate using the percentage of correct predictions, where a box is correctly predicted if its intersection-over-union with the ground-truth box is over 0.5.

For the referring expressions task, we use an ensemble RN50x16 and ViT-L/14@336 CLIP backbones. Following prior work [Subramanian et al. 2022; Y. Yao et al. 2021], we score the bounding box proposals of MAttNet [L. Yu et al. 2018].

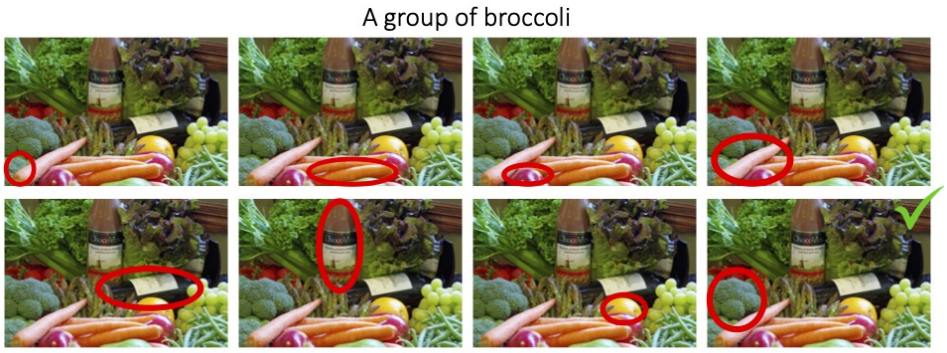


Figure 4.7: **Qualitative Evaluation on REC** on an image from RefCOCOg. CLIP correctly chooses the bottom right image. The circle is shown thicker than actually used for clarity (see the sup. matt. for the actual thickness).

Results. Using a red circle, we achieve state-of-the-art on most referring expressions comprehension baselines in the zero-shot setting, as shown in Table 4.5. Interestingly, this even outperforms ReCLIP [Subramanian et al. 2022], which is based on scoring image crops, followed by post-processing with manually designed relations rules. A red circle also outperforms Pseudo-Q on most benchmarks, even though Pseudo-Q explicitly trains for this task.

4.4.4 Model biases and ethics

While drawing circles on images can extract useful behaviors from a VLM for a wide variety of legitimate image analysis tasks, it can also extract unwanted ones. In particular, such a technique **must not be used for the analysis of sensitive data** in any circumstance.

To demonstrate this fact, in Fig. 4.8 we take a random image from COCO that contains a male-looking and a female-looking individual and zero-shot classify the image, as well as the image with circles over each individual, into 4 categories: male, female, missing person, and suspected murderer. While this leads to a correct resolution of the apparent gender of the annotated person, the annotated images are more likely to be classified as containing a missing person or a murderer. While we cannot know for certain, we hypothesize that this is due to the presence in the CLIP model training data of missing person reports, police footage, or similar, where people have been marked.

We further quantify these biases following [Agarwal et al. 2021], using face crops from FairFace [Karkkainen and Joo 2021] and person crops from COCO [T.-Y.

Model	Red Circle	**FairFace**	**COCO**
ViT-L/14@336px	✗	8.7%	38.4%
ViT-L/14@336px	✓	20.5% (+11.8%)	48.5% (+10.1%)
ViT-L/14	✗	10.2%	37.6%
ViT-L/14	✓	20.3% (+10.1%)	55.3% (+17.7%)
ViT-B/16	✗	11.5%	28.2%
ViT-B/16	✓	27.6% (+16.1%)	57.7% (+29.3%)
ViT-B/32	✗	9.0%	25.1%
ViT-B/32	✓	26.1% (+17.1%)	48.2% (+23.1%)
RN50x16	✗	17.0%	56.2%
RN50x16	✓	28.0% (+11.0%)	52.7% (-3.5%)

Table 4.6: **Classification rate into criminal categories.** We report ZS classification results in criminal categories on crops of faces from FairFace and persons from COCO, with and without red circles. In brackets is the absolute difference from the crop without a red circle.

Lin et al. 2014]. For FairFace, we classify 1000 random images into the original FairFace categories: *{White, Black, Indian, Latino, Middle Eastern, Southeast Asian, East Asian}, {man, woman}* using the template "*This is an image of a {race} {gender}*" for a total of 14 combinations. For COCO, we crop all bounding boxes for the class *person* from the validation set that have an area of at least 10% of the total area of the image, which comes down to 1352 crops. We use the template "*This is an image of a { }*", and the classes *{man, woman, person}*. For both FairFace and COCO, we add 3 criminal categories: *{criminal, thief, suspicious person}*, as in [Agarwal et al. 2021]. Finally, we zero-shot classify the original images and the images with circles. In Table 4.6 we present classification rates into criminal categories. We see that for all ViT encoders, the rate at which people are classified as criminals is significantly higher. This is problematic as such existing biases can lead to harmful consequences. Note that there are various limitations in this analysis, including the usage of binary gender.

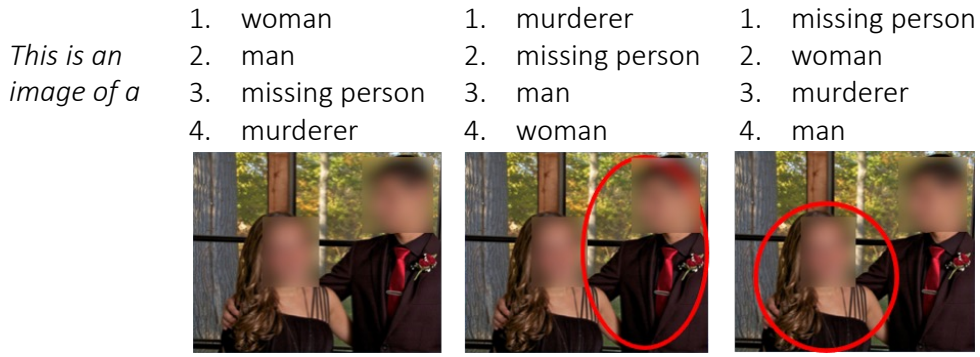


Figure 4.8: **Bias of CLIP.** We zero-shot classify a random COCO image with a male-looking and a female-looking individual, also adding a circle over each of them. We score the following sentences to the images: *This is an image of a {woman, man, missing person, suspected murderer }.* The apparent gender resolution is correct, but the circled images tend to be scored higher as missing or murderers. Blur added for privacy.

4.5 Conclusions

We have shown that visual prompt engineering via marking can extract useful behavior from VLMs such as CLIP in a zero-shot manner, achieving state-of-the-art zero-shot referring expression comprehension performance, and significantly outperforming traditional techniques like image cropping. Our analysis suggests that this behavior emerges because relevant samples of marking exist in the training data of the VLMs, but these samples are very rare. As a consequence, the behavior can only be learned by very large models trained on very large datasets. The analysis also shows that VLMs acquire undesirable behaviors too, where the mere addition of a red circle to an image increases the model's belief that the image has a negative connotation.

Ethics. We use the RefCOCO, RefCOCO+, MS-COCO, FairFace (YFCC100M), YFCC15M, CUB, SPair71k in a manner compatible with their terms. Some of these images may accidentally contain personal data (faces), but there is no extraction of personal or biometric information in this research. The only exception are the experiments in Section 4.4.4, where FairFace and MS-COCO and used to measure the bias in the off-the-shelf CLIP model used in the experiments. The FairFace dataset was collected specifically for this type of analysis. For further details on ethics, data protection, and copyright please see `https://www.robots.ox.ac.uk/~vedaldi/research/union/ethics.html`.

Acknowledgements. We thank Luke Melas-Kyriazi, Tim Franzmeyer, Rhydian Windsor and Bruno Korbar for proofreading A. Shtedritski is supported by EPSRC EP/S024050/1. A. Vedaldi and C. Rupprecht are supported by ERC-CoG UNION 101001212. C. Rupprecht is also partially supported by VisualAI EP/T028572/1.

4.6 Appendix

In this supplementary material, we provide more details about the datasets we use, implementation details and ablations, as well as further qualitative and quantitative evaluations.

4.6.1 Datasets

As noted in the main paper, we contribute additional annotations to the Spair71k dataset for some of our experiments. We start from their keypoint annotations, which have no keypoint name annotations in the original dataset. We then manually name all keypoints of the animal classes in Spair71k, as shown in Table 4.8. We purposefully leave out some point annotations:

- All animals have a left and right nostril annotated — we take the right one in all classes and annotate it as *nose*, and leave the left nostril out.
- All tails have point annotations at the start of the tail (attached to the body) and end of the tail. Because of the lack of words to precisely describe both points, we take the point *not* attached to the body and annotate it as *tail*, and leave the other one out.
- All ears have point annotations at the start of the ear (attached to the head), and at the pointy end. Because of the lack of words to precisely describe both points, we take the point *not* attached to the head and annotate it as *ear*, and leave the other one out.
- Birds have annotations for (i) foot, (ii) ankle, (iii) knee, which are often ambiguous and very close together. We only keep the *foot* annotation.

Method	Backbone	RefCOCO			RefCOCO+			RefCOCOg	
		Val	TestA	TestB	Val	TestA	TestB	Val	Test
ReCLIP	RN50×16	37.61	38.32	37.19	44.12	46.02	41.81	55.94	54.36
	ViT-B/32	40.69	43.98	37.55	45.00	48.15	41.65	55.25	54.35
	ViT-B/16	38.23	40.53	37.00	41.53	42.91	41.32	55.19	55.16
	ViT-L/14	34.40	33.52	34.35	37.86	37.53	37.70	53.82	52.25
	ViT-L/14@336px	35.90	37.72	35.66	40.06	42.49	39.07	54.25	53.92
	RN50×16,ViT-B/32	**41.96**	43.52	39.00	**47.44**	**50.11**	**43.93**	57.76	**57.15**
	RN50×16,ViT-B/1	39.94	41.61	38.71	45.06	47.17	43.63	57.93	56.85
	RN50×16,ViT-L/14	37.98	38.08	37.51	42.87	44.57	41.66	56.78	56.02
	RN50×16,ViT-L/14@336px	38.79	39.49	37.82	44.27	46.44	42.46	57.86	56.28
	ViT-B/32,ViT-B/16	41.34	**44.25**	38.55	45.20	48.01	43.36	57.37	56.52
	ViT-B/32,ViT-L/14	39.68	41.65	37.84	43.74	46.25	41.17	56.74	56.07
	ViT-B/32,ViT-L/14@336px	40.82	43.47	**39.22**	45.41	48.52	42.83	**58.09**	56.94
	ViT-B/16,ViT-L/14	37.69	38.29	37.53	40.87	42.07	40.93	56.35	55.76
	ViT-B/16,ViT-L/14@336px	39.18	41.01	38.35	42.81	44.32	42.07	57.82	56.21
	ViT-L/14,ViT-L/14@336px	35.47	36.26	35.70	39.52	40.69	38.70	54.51	54.04
Red Circle	RN50×16	45.52	52.99	38.59	49.98	57.55	42.11	53.94	54.35
	ViT-B/32	38.72	45.09	33.52	42.85	49.46	36.53	45.81	45.57
	ViT-B/16	45.30	52.70	36.51	49.39	57.67	40.60	53.72	53.26
	ViT-L/14	46.71	55.03	39.24	52.07	58.63	42.83	57.00	56.40
	ViT-L/14@336	48.27	56.44	39.71	53.59	59.99	43.28	**59.95**	58.51
	RN50×16, ViT-B/32	45.62	54.04	37.13	50.73	60.46	41.69	54.00	53.84
	RN50×16, ViT-B/16	**49.98**	57.15	38.04	52.98	61.95	42.99	56.01	55.78
	RN50×16, ViT-L/14	48.50	58.03	39.76	54.56	63.17	44.41	58.17	57.76
	RN50×16, ViT-L/14@336	49.84	**58.57**	39.96	55.28	**63.92**	**45.35**	59.40	**58.93**
	ViT-B/32,ViT-B/16	44.62	53.03	35.90	49.13	58.96	40.21	52.23	51.61
	ViT-B/32,ViT-L/14	47.19	56.27	38.14	52.75	62.07	42.69	56.66	55.54
	ViT-B/32,ViT-L/14@336px	48.59	58.05	38.69	54.61	63.45	43.28	57.80	57.48
	ViT-B/16,ViT-L/14	48.18	57.49	39.33	53.66	62.38	43.36	57.56	57.45
	ViT-B/16,ViT-L/14@336px	49.86	58.41	39.92	**55.35**	62.43	44.34	59.05	58.82
	ViT-L/14,ViT-L/14@336px	48.82	57.03	**40.35**	53.62	60.65	44.04	59.03	58.27

Table 4.7: Backbone ablation on **Referring Expressions Detection.** We compare CLIP backbones and their ensembles for ReCLIP [Subramanian et al. 2022] (without using relations resolution) and our Red Circle. The best and second best for each method are **bolded** and underlined, respectively.

4.6.2 Additional implementation details

Referring Expressions Detection.

Backbone. We base the evaluation of our method on ReCLIP [Subramanian et al. 2022], where an ensemble of two CLIP backbones is used — RN50x16 and ViT-B/32. We evaluated ReCLIP for all combinations of CLIP backbones in Table 4.7 and found that, on average, this is the highest-performing one. Similarly, for our method, we choose the ensemble of two backbones that lead to the highest performance — RN50x16 and ViT-L/14@336. Full comparison between the backbones can be found in Table 4.7.

Annotations. We experiment with different marker shapes, sizes, and colours, and present the results in Table 4.10. We find that, on average, a thin red circle leads to the best performance. We use an ensemble of the red circle annotation and two additional augmentations — blurring and gray-scaling the outside of the circle, for a total of three images per annotation, as shown. These augmentations were inspired by examples in YFCC15M we discovered that were annotated like that. We found that adding augmentations improves overall results. However, we did not explore including augmentations beyond these. We ablate these choices in Table 4.9.

Additional details. We augment the text queries by prepending *"This is"*. When subtracting the average with respect to other referring expressions, we use $Q = 500$ randomly sampled expressions.

Keypoint tasks

Backbone. We evaluate different backbones in Table 3 in the main paper and find that ViT-L/14@336 performs the best.

Annotations. We show examples of the markers we use in Fig. 4 in the main paper . We compare a large range of sizes and colors, as shown in Table 2 in the main paper. We find that a circle is the best marker, and drawing a cross over the point of interest is the worst. The best-performing marker out of all is a red circle, which is the one we end up using. In Fig. 4.10 we show a more detailed comparison of different colors, diameters, and thicknesses when using a circle annotation. We see that a thin red circle is the best-performing marker. We show what that circle

looks like on an image in Fig. 4.11. Given this, we draw red circles over the images, with radius $r = 0.06H$ and thickness $t = 0.01H$, where H is the shorter side of the image. For the backbone we use, where the input size has $H = 336$px, this becomes $r = 20$px and $t = 3$px.

Additional details. For the keypoint localization task, we set $M = 30$, for a total of $30 \times 30 = 900$ query locations before applying the pseudo mask. The templates we use are *"This is the {part} of a bird"* for CUB and *"This image shows the {part} of the {animal}"* for SPair71k. We use a temperature parameter $\tau = \frac{1}{150}$.

4.6.3 Qualitative evaluations

We present qualitative evaluations on naming keypoints in Figs. 4.14 and 4.15, keypoint localization in Figs. 4.11 and 4.12 and referring expressions comprehension in Fig. 4.13.

Part No	Bird	Cat	Cow	Dog	Horse	Sheep
0	crown	—	—	—	—	—
1	right wing	—	—	—	—	—
2	left wing	right ear	right ear	right ear	right ear	right ear
3	beak	left ear	left ear	left ear	left ear	left ear
4	—	right eye	right eye	right eye	right eye	right eye
5	—	left eye	left eye	left eye	left eye	left eye
6	forehead	nose	nose	nose	nose	nose
7	right eye	—	—	forehead	—	—
8	left eye	mouth	mouth	mouth	mouth	mouth
9	nape	front right paw	front right hoof	front right paw	forehead	front right hoof
10	right foot	front left paw	front left hoof	front left paw	front right hoof	front left hoof
11	left foot	hind right paw	hind right hoof	hind right paw	front left hoof	hind left hoof
12	—	hind left paw	hind left hoof	hind left paw	hind right hoof	hind right hoof
13	tail	tail	tail	tail	hind left hoof	tail
14	—	—	—	—	tail	—
15	—	—	front right knee	neck	—	front right knee
16	—	—	front left knee	—	front right knee	front left knee
17	—	—	hind right knee	—	front left knee	hind right knee
18	—	—	hind left knee	—	hind right knee	hind left knee
19	—	—	right horn	—	hind left knee	right horn
20	—	—	left horn	—	—	—

Table 4.8: **Part names for keypoint annotations of the SPair71k dataset**. Part No is the part number in the SPair71k annotations. Some parts are annotated inconsistently in the original annotations, e.g. "tail" is part number 10 for the "horse" class, but part number 9 for all other animal classes.



Figure 4.9: **Annotations for Referring Expressions Detection.** Here we show the annotation types we consider. A: original bounding box annotation. B: Red Circle. C: Red Circle + Blur outside. D: Red Circle + Gray outside. In our experiments, we use an ensemble of B, C and D unless stated otherwise.

Component			RefCOCO			RefCOCO+			RefCOCOg	
Red Circle	Subtract	Ensemble	Val	TestA	TestB	Val	TestA	TestB	Val	Test
✓	✗	✗	42.01	48.58	36.90	47.55	53.56	41.05	50.84	51.47
✓	✓	✗	43.67	50.20	38.59	48.98	54.70	43.06	54.29	52.98
✓	✓	✓	**49.84**	**58.57**	**39.96**	**55.28**	**63.92**	**45.35**	**59.40**	**58.93**

Table 4.9: **Ablation study.** We ablate subtracting the mean wrt negative queries and ensembling different marker types (red circle + red circle and blur outside + red circle and grey outside). Here we use RN50×16 and ViT-L/14@336px backbones and a red circle with the optimal size described in Table 4.10

Annotation Type			RefCOCO			RefCOCO+			RefCOCOg	
Shape	Color	Size	Val	TestA	TestB	Val	TestA	TestB	Val	Test
Circle	Red	1	38.7	<u>45.1</u>	<u>34.0</u>	<u>44.4</u>	**50.0**	<u>39.1</u>	48.1	**50.0**
Circle	Red	2	32.2	35.9	29.1	37.6	40.9	33.5	45.3	46.4
Circle	Red	4	37.4	43.6	31.5	43.3	47.8	37.3	43.7	48.0
Circle	Red	8	36.3	42.6	31.3	42.1	47.3	36.3	45.2	45.4
Rectangle	Red	1	35.1	38.3	33.5	39.2	41.4	37.3	44.3	43.4
Rectangle	Red	2	35.1	38.3	33.2	39.1	41.8	37.3	44.8	44.1
Rectangle	Red	4	34.1	37.8	32.3	39.0	41.3	36.5	43.7	44.1
Rectangle	Red	8	33.7	37.6	32.7	37.9	40.3	34.9	41.1	40.1
Circle	Green	1	**39.3**	**45.4**	**34.8**	43.8	<u>49.9</u>	38.1	47.2	47.4
Circle	Purple	1	<u>38.9</u>	44.8	34.0	**44.5**	49.4	**39.2**	**49.5**	<u>49.2</u>
Circle	Blue	1	37.7	44.9	33.5	43.4	49.1	37.3	48.2	48.3
Circle	Yellow	1	38.5	44.1	34.6	43.7	49.0	38.9	<u>48.6</u>	48.1

Table 4.10: **Comparison of different sizes, shapes, colors.** A unit size of 1 corresponds to 0.5% of the larger side of the image, which is 1 pixel for an image of size 224. Here we do *not* use ensembling and subtraction of the mean wrt other queries in order to evaluate the effectiveness of different markers themselves. The best and second best are **bolded** and <u>underlined</u>, respectively.

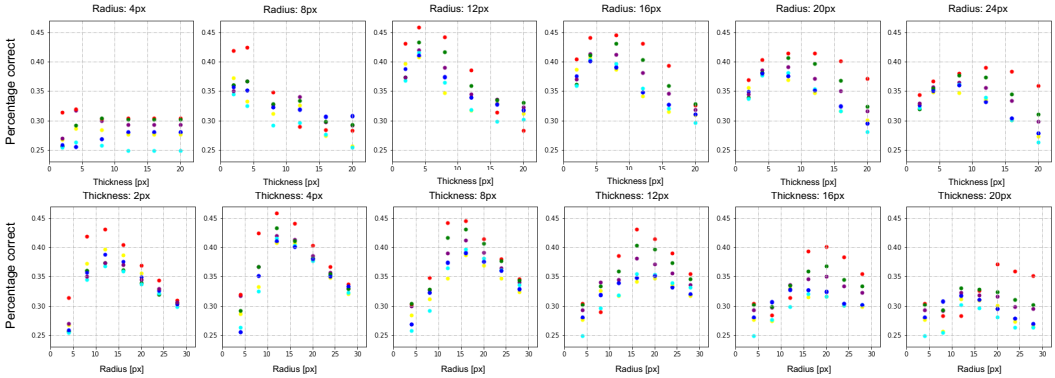


Figure 4.10: **Ablation of circle sizes and colours for keypoint matching.** We present results on the CUB dataset when varying the diameter and thickness of the annotations. The presented numbers are for text-to-image matching. The best performing annotation has a radius of 12px and thickness of 4px. The colour of the dots on the scatter plot illustrates the colour of the annotation — red, green, blue purple, yellow, cyan.

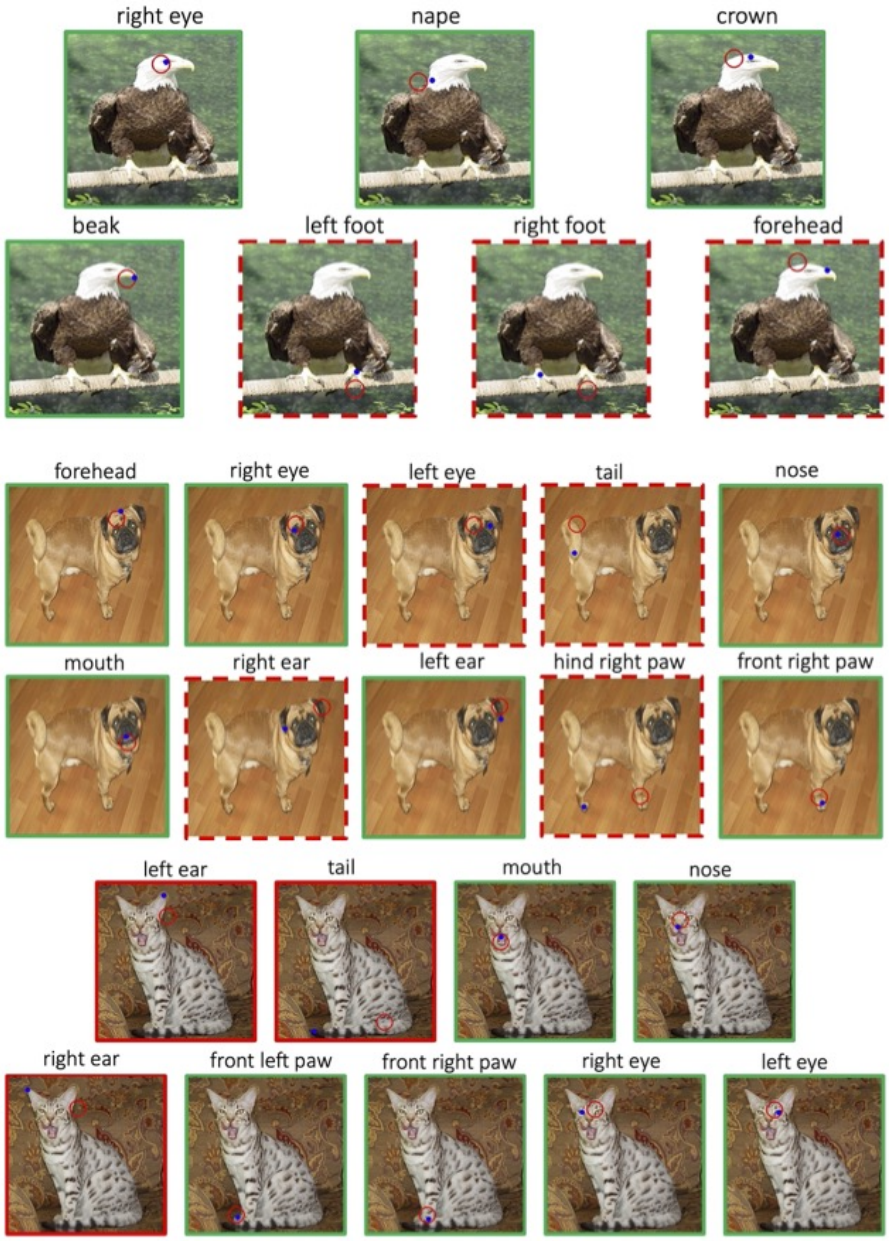


Figure 4.11: **Qualitative evaluation of keypoint localization on SPair71k.** We show all keypoint names for the images and color code in green and red (dashed) the correct and wrong localizations according to PCK with $\alpha = 0.1$. The red circle is the marker we use and the blue dot is the ground truth location.



Figure 4.12: **Qualitative evaluation of keypoint localization on SPair71k.** We show all keypoint names for the images and color code in green and red (dashed) the correct and wrong localizations according to PCK with $\alpha = 0.1$. The red circle is the marker we use and the blue dot is the ground truth location.

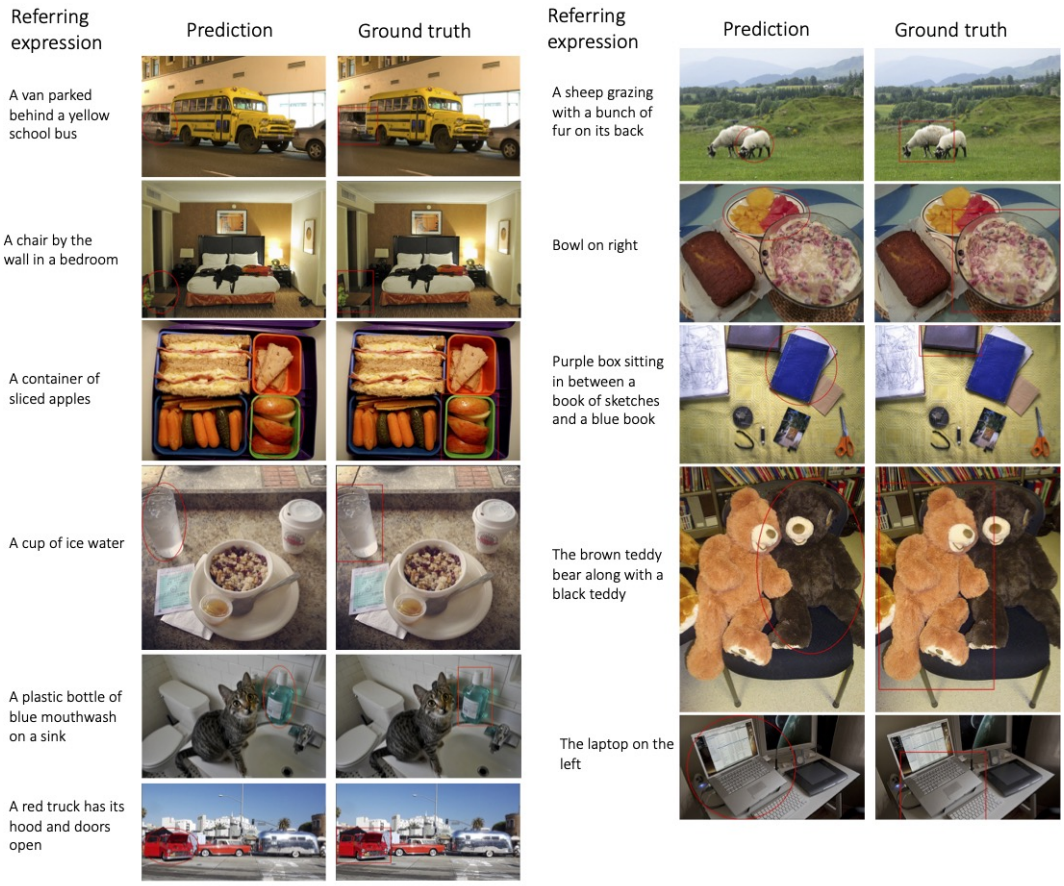


Figure 4.13: **Qualitative results on REC on the RefCOCOg dataset.** Left: correct predictions. Right: wrong predictions. The last row on the right shows an example where the ground-truth bounding box is wrong.

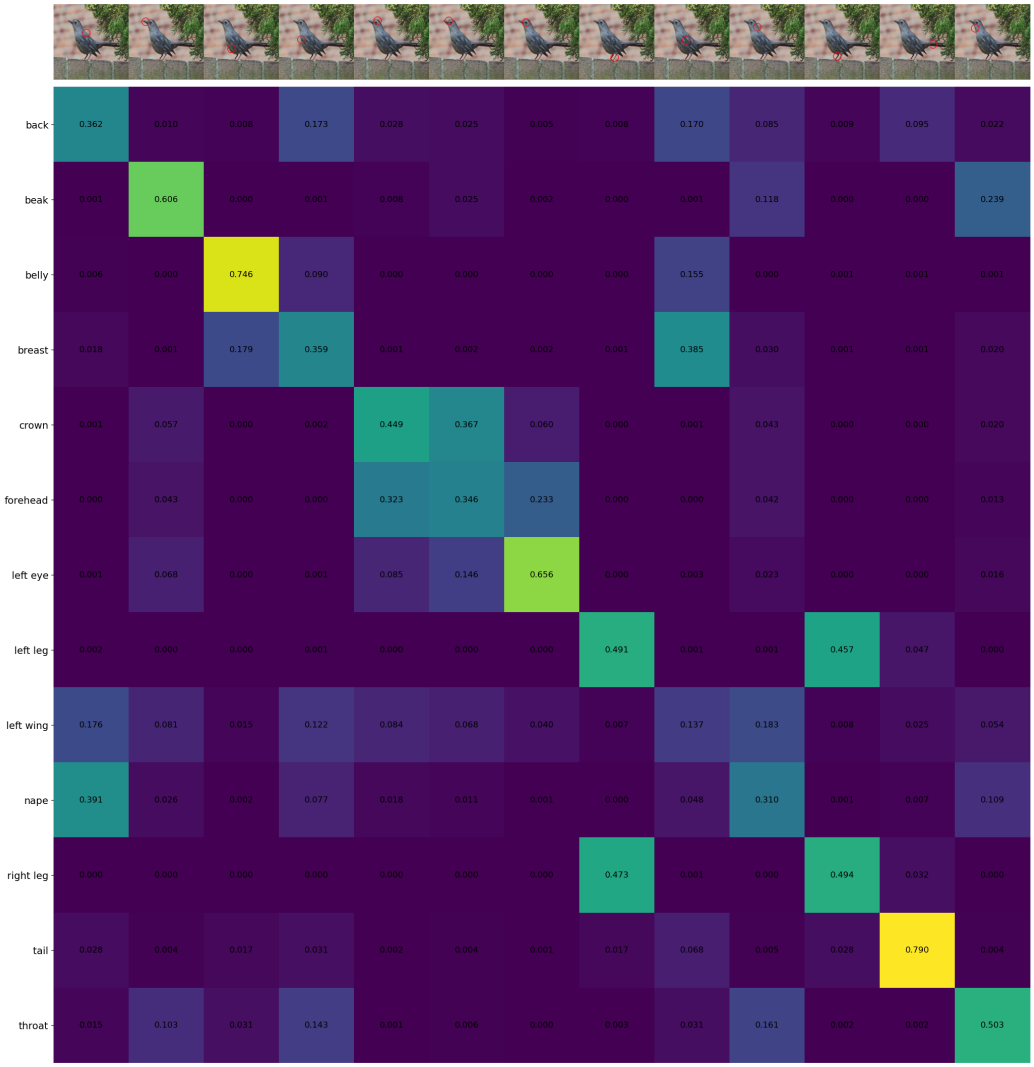


Figure 4.14: **Naming keypoints.**Normalized cost matrix for an image from CUB

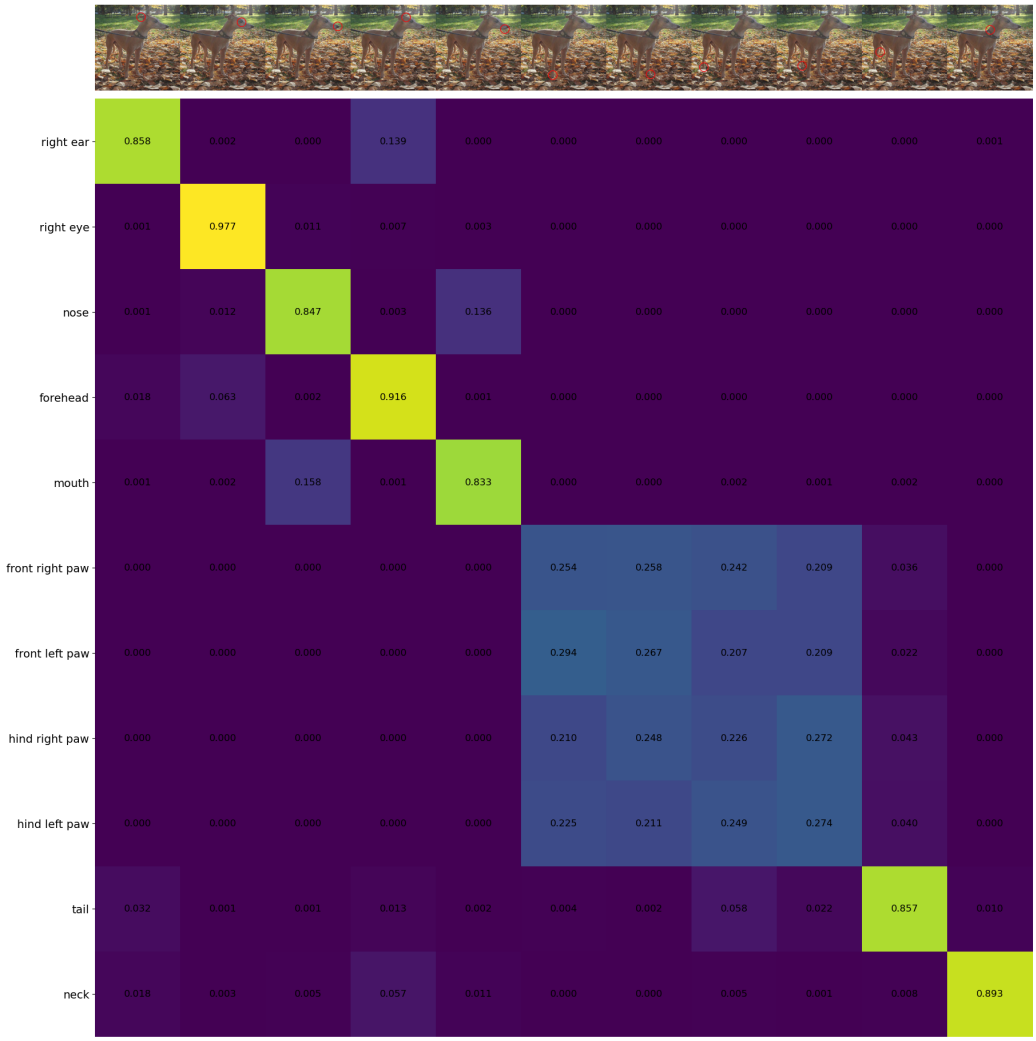


Figure 4.15: **Naming keypoints.** Normalized cost matrix for an image from SPair71k

Chapter 5

Highlight: Learning Visual Prompts for Vision-Language Models

The was presented at the CVPR 2025 EVAL-FoMo workshop.

Highlight: Learning Visual Prompts for Vision-Language Models

Jana Zeller, Aleksandar Shtedritski, Christian Rupprecht

Visual Geometry Group, University of Oxford

{jana, suny, chrisr}@robots.ox.ac.uk

jana-z.github.io/highlight-website

September 29, 2025

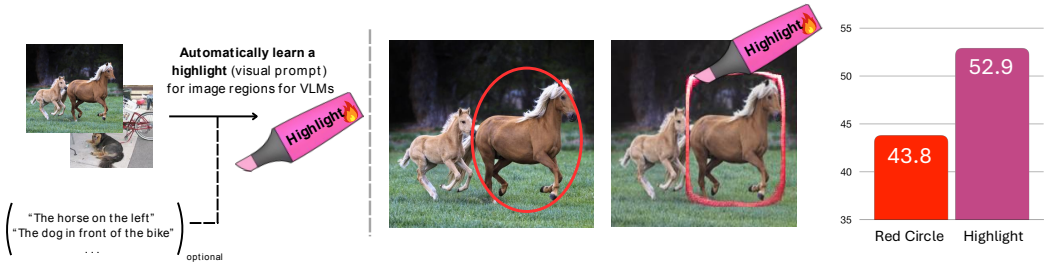


Figure 5.1: **Highlight automatically learns to visually prompt images.** Left: Given an image collection and, optionally, descriptions for image regions, Highlight learns a visual marker that is salient for VLMs. Right: A red circle vs the visual marker learnt by Highlight. Highlight outperforms a red circle by 15% on the RefCOCO, RefCOCO+ and RefCOCOg datasets, on average.

Abstract

Large-scale Vision-Language Models, such as CLIP, demonstrate impressive capabilities and have multiple applications, from text-to-image generation to zero-shot classification. Recent work has suggested that visual prompts, such as a red circle, can steer the vision encoder to the circled region. While such vision prompts have now been used in various applications, they might be model-specific and depend on the model learning these behaviours from its training data. Discovering and evaluating various prompts might not be feasible given different models, tasks, and datasets. In this paper, we propose *Highlight*, a method to learn a visual prompt that highlights a region in an image or refines a manually engineered visual

prompt. Using our framework, we can learn to highlight in a supervised way using a dataset of text-image region pairs or in an unsupervised way using synthetic captions or images only. Interestingly, we find that *Highlight* "discovers" visual prompts which are strikingly similar to manually-engineered ones, like red contours. *Highlight* outperforms other visual prompts, prompt learning approaches, and compute-intensive methods that use ensembles of multiple models and visual prompts.

5.1 Introduction

Vision Language Models (VLMs) have shown outstanding performance across a range of tasks: For instance, CLIP [Radford et al. 2021] can perform zero-shot classification in a number of domains, and Flamingo [Alayrac et al. 2022] can count objects within an image, even though these tasks were not explicitly part of their training. Instead, it is hypothesised that these abilities *emerge* from the diverse large-scale training data. However, since the alt texts in pretraining image-text datasets summarise global image descriptions, these models have limited capabilities for localising objects. This limited spatial awareness hinders the usage of VLMs in various potential applications, such as autonomous driving [Wen et al. 2023], robotics [Brohan et al. 2023; Driess et al. 2023; J. Gao et al. 2023], and assistive technologies [Zongming Yang et al. 2022].

A promising new approach, *visual prompting* [Junda Wu et al. 2024], has emerged to address these limitations. By introducing visual markers, such as red circles, to highlight target objects, pretrained foundation models can reason about localised image regions. Visual prompting for VLMs has found multiple applications, such as medical image analysis [Denner et al. 2024], trajectory selection [Daeun Song et al. 2024], and audio descriptions generation [Junyu Xie et al. 2024]. However, despite the different domains of application of visual prompting and the different visual encoders used, these methods rely on simple markers, such as a red circle, which have only been found to perform well empirically.

It has been suggested that VLMs learn to pay attention to such visual markers from their training data, *e.g.* [Shtedritski et al. 2023b] discover images annotated with red circles in YFCC100M [Thomee et al. 2016], which is part of CLIP's

pretraining data. However, whether a red circle is the prompt that best elicits these emergent behaviours in VLMs is unclear. Furthermore, there might not be one optimal prompt per model—different domains in the training data might be annotated differently because of arbitrary conventions in fields like medicine or science. Thus, it becomes unfeasible to manually guess and evaluate visual prompts for every new VLM and domain to which it is applied.

In this paper, we introduce *Highlight*, a method to automatically learn a visual prompt that highlights regions for a VLM. Highlight visually prompts an image in a differentiable way by predicting both the shape and the colour of the visual marker. To supervise Highlight, first, we consider the case when annotated text-image region pairs are available. We show that in that case, we can use CLIP's multimodal contrastive objective to match prompted images with text, learning a state-of-the-art visual marker. However, this requires manual annotations. We thus introduce an unsupervised version of Highlight that is trained with images only and does not require manual supervision. Our key insight is that we can replace the textual description of a region in the image-text contrastive objective either with a synthetically generated caption for the image patch or with a visual description of the region, *e.g.* a crop. For example, we can learn to match highlighted images with a crop of the image. We find that the visual marker we learn in that way outperforms manually engineered visual markers. Interestingly, however, the learnt visual marker looks strikingly similar to the manually engineered visual markers, such as a red circle or rectangle.

5.2 Related Work

Visual prompt enginering, as introduced by [Shtedritski et al. 2023b], uses visual markers in the input image to change the output of a VLM. For example, [Shtedritski et al. 2023b] draw red circles around objects of interest in order to make the VLM focus on these specific areas, whereas [Lingfeng Yang et al. 2024] propose pixel-wise outlines of objects in the image. [Jinheng Xie et al. 2024] proposes to optimise the position of a visual marker to discover objects within an image. Both [Lingfeng Yang et al. 2024] and [Shtedritski et al. 2023b] hypothesise that their proposed prompts are salient for the model because similar visual markers are

present in the training data. Instead, [Rezaei et al. 2024] optimise the colours within a fixed mask (*e.g.* a circle) to draw the attention of transformers towards the prompted area. However, the method of [Rezaei et al. 2024] only applies to transformers and requires access to the internal features of the ViT. Instead, Highlight treats the vision model as a black box and is thus architecture-agnostic. Furthermore, our framework can accommodate various forms of supervision, such as using image-text pairs, which allows learning a *supervised* visual prompt.

Referring Expression Comprehension (REC) is the task of localising an object in an image corresponding to a query textual description. Many REC methods start with object proposals, for example, generated with Faster-RCNN [Shaoqing Ren et al. 2015], and then learn to score them [Hu et al. 2016; R. Luo and Shakhnarovich 2017; Yongfei Liu et al. 2020b; Sibei Yang et al. 2019b; Peng Wang et al. 2019b]. Past research focuses on models specifically trained to solve the REC task. This includes combining pretrained vision and language models [J. Mao et al. 2016; L. Yu et al. 2016; Nagaraja et al. 2016; Hu et al. 2016; R. Luo and Shakhnarovich 2017], learning a joint-embedding space [L. Wang et al. 2016; J. Liu et al. 2017; K. Chen et al. 2017], graph-based models [Yongfei Liu et al. 2020b; Peng Wang et al. 2019b; Sibei Yang et al. 2019b], that yield better interpretability, and more recently transformer based architectures [Kamath et al. 2021; W. Kim et al. 2021; Jiajun Deng et al. 2021]. However, most of these rely on large amounts of training data, which include images, referring expressions, and the corresponding bounding boxes.

Unsupervised Referring Expression Comprehension has recently been enabled with the introduction of large pretrained models, such as CLIP [Radford et al. 2021], which allow solving REC in a *zero-shot manner* [Shtedritski et al. 2023b; Y. Yao et al. 2024; Subramanian et al. 2022; Kirillov et al. 2023a]. ReCLIP [Subramanian et al. 2022] ranks object crops using CLIP before a post-processing step that considers relations between objects. CPT [Y. Yao et al. 2021] colours objects and uses a pre-trained captioning model [P. Zhang et al. 2021] to predict which colour corresponds to the referring expression. More recently, unsupervised methods overlay visual prompts on objects [Shtedritski et al. 2023b; Lingfeng Yang et al. 2024; Rezaei et al. 2024]. However, most of these methods use ensembles of visual prompts and/or vision-language models to achieve competitive performance. In-

stead, our method automatically discovers a visual prompt that outperforms such ensembles, making it several times more compute-efficient.

Visual patches can be optimised to steer the output of a vision model. [Papernot et al. 2016] demonstrate that altering only a few pixel values can divert the model's output towards an adversarial output. [Schlarmann and Hein 2023] recently transferred this work to VLMs, where the objective is to move the embedding of an adversarial image towards some harmful textual embedding. Their work differs from ours in requiring the adversarial changes to be invisible to humans. More closely related to our work, *adversarial patches* [T. B. Brown et al. 2017] restrict the alteration to a region. Our method can be seen as an extension of this research. However, unlike adversarial patches, we do not aim to mislead the model's output towards a harmful or incorrect outcome. Instead, we focus on optimising visual prompts to improve task performance.

5.3 Method

We aim to learn a *visual prompt* for a VLM that focuses the visual representation on a particular object in an image. To solve a prediction task, models such as CLIP jointly encode images and text and learn to match them. An input to such a model is an image $I \in \mathbb{R}^{3\times H\times W}$ and text T, and the output measures the similarity between both inputs.

5.3.1 Visual prompting

A model such as CLIP can be used to find an object referred to by a query text prompt T by visually prompting an image. First, a visual prompting function $\mathcal{F}(I, u)$ is used to highlight one object proposal u in the image I, where u is the location of an object in the image, which can be, *e.g.* bounding box coordinates $u = (x_c, y_c, x_w, y_h)$. For a set of possible object locations U, the location $\hat{u}$ in the image that is best described by the referring expression T is then given by

$$\hat{u} = \arg\max_{u\in U} s(\mathcal{F}(I, u), T), \tag{5.1}$$

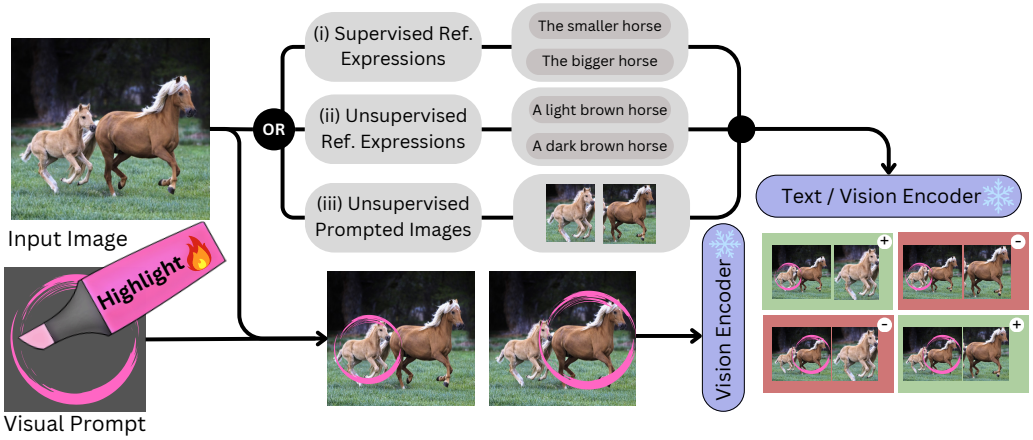


Figure 5.2: **Overview of Highlight.** Highlight generates a visual prompt, which is then alpha-blended with object proposals in the image. To learn the visual prompt, we construct positive and negative pairs for each object using: (i) supervised text and bounding box pairs (e.g. from RefCOCO), (ii) unsupervised text and bounding box pairs obtained by captioning the bounding box, (iii) a visual representation of the object, *e.g.* a crop of the bounding box as in the Figure, or the original image, visually prompted with a red circle. Here, (ii) and (iii) are unsupervised in that they do not require manual text-image region annotations. The CLIP image and text encoder are kept frozen.

where $s(I, T)$ is the compatibility score between an image and text, *e.g.* as measured by a VLM. To implement the visual prompting function $\mathcal{F}$, prior work crop the image [Subramanian et al. 2022], overlay red circles [Shtedritski et al. 2023b], or generate pixel-wise outlines [Lingfeng Yang et al. 2024]. In this section, we show how we can learn such a prompt.

5.3.2 Overlaying a visual prompt.

We decompose a visual prompt as $\mathcal{F} = \mathcal{F}_\alpha \otimes \mathcal{F}_{RGB}$, where $\mathcal{F}_\alpha \in \mathbb{R}^{H \times W}$ is an alpha channel and $\mathcal{F}_{RGB} \in \mathbb{R}^{3 \times H \times W}$ is a prompting image. To visually prompt an image I, we then have

$$\mathcal{F}(I, u) = \begin{cases} \mathcal{F}_\alpha(v) \cdot \mathcal{F}_{RGB}(v) + \Big(1 - \mathcal{F}_\alpha(v)\Big) \cdot I(v) & \text{if } v \in u, \\ I(v) & \text{else,} \end{cases}$$

performing alpha blending for all locations $v \in \mathbb{R}^2$ that are within the region u.

5.3.3 Highlight: Learning a Visual Prompt

Instead of using a manually engineered visual prompt, our method Highlight learns a visual prompting function $\mathcal{F}$ to highlight an object. A schematic of Highlight is shown in Fig. 5.2

Learning a visual prompt. We notice that the region selection in Eq. (5.1) resembles classification, where the most compatible region to the text query is selected. Given a dataset which consists of text queries T, where for each T we have a positive and several negative regions (that are *not* described by T), u and U_n in an image I, we can directly learn $\mathcal{F}$ such that we fulfill Eq. (5.1):

$$\mathcal{L}_s(I, u, U_n, T) = \mathcal{L}_{\text{InfoNCE}}\left(T, \mathcal{F}(I, u), \mathcal{F}(I, u_n)_{u_n \in U_n}\right), \tag{5.2}$$

where

$$\mathcal{L}_{\text{InfoNCE}}(x, x^+, X^-) = -\log\left(\frac{\exp\left(\text{sim}(x, x^+)/\tau\right)}{\exp\left(\text{sim}(x, x^+)/\tau\right) + \sum_{x^- \in X^-} \exp\left(\text{sim}(x, x^-)/\tau\right)}\right),$$

is the InfoNCE loss [Oord et al. 2018] . Here, τ is a temperature parameter, and

$$\text{sim}(I, T) = \frac{\langle \Phi_I(I), \Phi_T(T) \rangle}{\|\Phi_I(I)\| \|\Phi_T(T)\|} \tag{5.3}$$

is the cosine similarity between the embeddings of the image and text using the CLIP image and text encoders, Φ_I and Φ_T. Intuitively, minimising Eq. (5.2) will give us a prompting function $\mathcal{F}$ that focuses the global image embedding on the highlighted prompted object, making the image's embedding similar to the textual description of the prompted object while also making the embedding dissimilar to other textual descriptions of other objects in the image.

Unsupervised visual prompts with synthetic text-image pairs. The loss $\mathcal{L}_s$ in Eq. (5.2) is supervised in that it requires manual annotations in the form of region-specific captions, which can be limiting and expensive to acquire. Thus, we can augment this method by using synthetic captions $\tilde{T}_u$ (as recently used to train CLIP [Nguyen et al. 2024; Hammoud et al. 2024]) generated by BLIP-2

[Junnan Li et al. 2023] for a given crop. With this, we can also generate captions for the negative locations, giving us additional negative image-text pairs T_{u_n}. The unsupervised loss becomes:

$$\begin{aligned}\mathcal{L}_u(I, u, U_n, \tilde{T}) = \mathcal{L}_{\text{InfoNCE}}(\mathcal{F}(I, u), \tilde{T}_u, \tilde{T}_{u_n, u_n \in U_n}) \\ + \mathcal{L}_{\text{InfoNCE}}(\tilde{T}_u, \mathcal{F}(I, u), \mathcal{F}(I, u_n)_{u_n \in U_n})\end{aligned} \tag{5.4}$$

Unsupervised visual prompts with image-image pairs. The naive, unsupervised objective proposed above suffers from two main drawbacks. First, obtaining the captions requires a captioning VLM, which might not be available for some domains and introduces additional computational overhead. Furthermore, the descriptions generated by the captioner might be *ambiguous*. To see why that is the case, consider the example in Fig. 5.2. The supervised referring expressions in datasets such as RefCOCO [Kazemzadeh et al. 2014] are explicitly designed such that the objects can be uniquely identified from their descriptions. However, this can not enforced on the captioning VLM, which has no global information about the scene. Instead of the captions {"A light brown horse" and "A dark brown horse"} (as in Fig. 5.2), the captioner might output {"A horse" and "A horse"}, from which the objects can not be disambiguated, leading to a sub-optimal learning signal.

Instead of matching a visually prompted image to a textual description of it, we propose to match it with the visual content in the prompt instead. To this end, we replace each description T_u of the region u in Eq. (5.4), with a visual representation of the region u. We denote this with $\mathcal{F}^*$, where $\mathcal{F}^*(I, u)$ might be any visual prompting function, such as a crop of the region u or the image, prompted with a red circle in u. This gives us the unsupervised objective:

$$\begin{aligned}&\mathcal{L}_{u,i2i}(I, u, U_n) \\ &= \mathcal{L}_{\text{InfoNCE}}(\mathcal{F}(I, u), \mathcal{F}^*(I, u), \mathcal{F}^*(I, u_n)_{u_n \in U_n}) \\ &+ \mathcal{L}_{\text{InfoNCE}}(\mathcal{F}^*(I, u), \mathcal{F}(I, u), \mathcal{F}(I, u_n)_{u_n \in U_n}).\end{aligned} \tag{5.5}$$

Visibility loss. We empirically observe that learning the alpha channel is sometimes unstable, and the model converges to a zero alpha channel, which results in poor performance. To tackle this, we simply penalise low average values of $\mathcal{F}_\alpha$:

$$\mathcal{L}_v = \max(0, \tau - \frac{1}{HW} \sum_{x,y} \mathcal{F}_\alpha(x, y)), \tag{5.6}$$

where τ is a threshold which controls when the loss turns off.

Training formulation. We freeze the CLIP text and image encoders and only train the prompt function $\mathcal{F}$, which we parametrise with a neural network. Given a dataset $\mathcal{D}$ of images and object proposals, our loss is

$$\mathcal{L} = \frac{1}{|\mathcal{D}|} \sum_{i,u,U_n \in \mathcal{D}} \beta \mathcal{L}_u(i, u, U_n) + \gamma \mathcal{L}_v,$$

where β and γ are hyperparameters. In practice, given a set of object proposals U, we can select any $u \in U$ and define the set of negative proposals as $U_n = U \setminus u$. If region-text pairs are available, we can swap $\mathcal{L}_u(i, u, U_n)$ for the supervised loss $\mathcal{L}_s(i, u, U_n, T)$.

Pretraining We found that our unsupervised method, especially when trained with image-image pairs, performs better when $\mathcal{F}$ has been initialised with a manually engineered visual marker, *e.g.* a red circle. Given such a marker, we pre-train $\mathcal{F}_{RGB}$ with mean squared error loss and $\mathcal{F}_\alpha$ with binary cross entropy loss.

5.4 Experiments

In this section, we describe the datasets used for our experiments and the implementation details of our method. After that, we evaluate Highlight against competing methods and ablate the different components we propose.

5.4.1 Datasets

We evaluate Highlight on Referring Expressions Comprehension. We use the RefCOCO [L. Yu et al. 2016], RefCOCO+ [L. Yu et al. 2016], and RefCOCOg [J. Mao et al. 2016] datasets, all of which consist of images from the MS-COCO

Table 5.1: **Performance on Referring Expressions Comprehension.** Our supervised prompt outperforms prior works that use ensembles of several models and prompts, which require up to 6x more forward passes. Our unsupervised prompt outperforms all other single-prompts methods by 4.8% on average. † as reported in [Shtedritski et al. 2023b]. ‡ as reported in [Jinheng Xie et al. 2024]. 🧍 contains expressions referring to people, 🪑 for objects. **Sup** indicates whether a method was supervised or not. The best result is marked in **bold**.

Method	Sup	Ensemble	RefCOCO			RefCOCO+			RefCOCOg		Avg
			Val (🧍, 🪑)	TestA (🧍)	TestB (🪑)	Val (🧍, 🪑)	TestA (🧍)	TestB (🪑)	Test (🧍, 🪑)	Val (🧍, 🪑)	
Crop	✗	✗	27.9	29.0	29.3	32.6	33.8	33.0	45.8	47.0	34.8
Reverse Blur	✗	✗	32.2	38.1	29.3	36.7	42.5	31.1	42.3	42.3	36.8
Red Circle	✗	✗	39.8	47.1	33.2	44.4	51.5	38.2	48.4	48.0	43.8
Highlight (Unsup., synth captions)	✗	✗	41.4	48.1	**36.1**	47.0	52.8	**41.0**	48.2	**48.2**	45.4
Highlight (Unsup., images only)	✗	✗	**42.1**	**49.3**	**36.1**	**47.6**	54.6	40.7	**48.5**	**48.2**	**45.9**
ReCLIP†	✗	✓	45.8	46.1	**47.1**	47.9	50.1	45.1	59.3	**59.0**	50.1
RedCircle†	✗	✓	**49.8**	**58.6**	39.9	**55.3**	**63.9**	45.4	**59.4**	58.9	**53.9**
Tune-An-Ellipse‡	✗	✓	**49.8**	58.0	40.9	55.1	62.8	**45.7**	58.4	58.7	53.7
Highlight (Supervised)	✓	✗	49.3	57.7	40.6	54.8	62.5	45.1	56.5	56.6	52.9

Table 5.2: **Evaluating the importance of optimising the prompt shape.** We fix the prompt shape of Highlight (unsupervised) to a circle and only learn its RGB values, which is fairly comparable to Guided Attention [Rezaei et al. 2024]. † as reported in [Rezaei et al. 2024]. The best result is marked in **bold**, and the second best underlined. Here we evaluate using the ground-truth proposals from RefCOCO, following in [Rezaei et al. 2024].

Method	Shape	RefCOCO			RefCOCO+			RefCOCOg		Avg
		Val (🧍, 🪑)	TestA (🧍)	TestB (🪑)	Val (🧍, 🪑)	TestA (🧍)	TestB (🪑)	Test (🧍, 🪑)	Val (🧍, 🪑)	
Guided Attention†	circle	46.7	49.5	43.9	48.1	51.3	46.2	50.4	49.0	48.1
Highlight (RGB only)	circle	61.0	68.1	54.0	65.7	72.5	60.0	74.3	73.5	66.1
Highlight	learnt	61.5	67.9	54.8	66.3	72.3	60.3	74.9	73.4	66.4

dataset [T.-Y. Lin et al. 2014] together with expressions that refer to a *unique* object in the image, which are also annotated with a bounding box. The unique expression-image region correspondence satisfies our condition in Eq. (5.2), where locations outside the bounding must *not* be described by the referring expression. RefCOCO+ only contains appearance-based expressions, whereas RefCOCO and RefCOCOg contain relation-based expressions (e.g., containing the words left/-closer/bigger). The test sets of RefCOCO and RefCOCO+ are split into two parts, where "TestA" and "TestB" contain only people and non-people, respectively. Following prior work [Subramanian et al. 2022; Y. Yao et al. 2021; Shtedritski et al. 2023b], we use the bounding box proposals of MAttNet [L. Yu et al. 2018]. For our unsupervised losses, we remove proposals with bounding box sizes of less than 50px. We train on all RefCOCO datasets combined, making sure to exclude any images that appear in the test sets (e.g. we found that some training images of RefCOCOg might appear as test images in RefCOCO+).

Method	RefCOCO	RefCOCO+	RefCOCOg
Crop	28.8	33.2	46.4
Highlight + Crop	41.7	47.0	47.6
Blur	33.2	36.8	42.3
Highlight + Blur	42.5	47.6	48.4
RedCircle	40.0	44.7	48.2
Highlight + RedCircle	40.8	45.9	47.7

Table 5.3: **Comparing different targets $\mathcal{F}^*$.** We compare different baseline visual prompts, with Highlight when trained with that visual prompt for its target $\mathcal{F}^*$. We see the biggest gains when using Crop, but the best performance when using Reverse Blur.

5.4.2 Implementation details

Architecture. While one can optimise a visual prompt directly in pixel space, the limited number of learnable parameters in pixel space hinders the sharing of spatial patterns between pixels, leading to less efficient optimisation [Krull et al. 2019]. To generate the visual prompt, we, therefore, use a Coordinate Network [Stanley 2007], which, given the (x, y) coordinates of a pixel location in the visual prompt, generates the RGB and α values for that location. Therefore, Coordinate Networks can be thought of as 2D Nerfs [Mildenhall et al. 2021]. The network generates a fixed-sized prompt, which is afterwards scaled to fit the size of the object proposal that should be prompted. We use a Coordinate Network with eight layers and a hidden dimension of 256. We add a sinusoidal positional encoding to the input and apply a sigmoid function to the outputted α mask.

Additional details. We use a linear annealing schedule with a 10% warmup and cooldown for all pretrained Highlights and pre-train them to output a red circle for 2,000 steps unless mentioned otherwise. For the models that were not pretrained, we empirically observed exploding gradients when utilising a scheduler. Therefore, we do not use a scheduler for these runs. Similarly to prior work, we use a CLIP ViT-L/14 in most experiments unless stated otherwise. The hyperparameters we used in different settings can be found in the Appendix.

5.4.3 Results

Supervised Highlight. In Table 5.1, we compare Highlight to other visual prompting methods. Since our framework is the first to allow supervised training of a visual prompt, we compare to ensemble methods, which usually perform

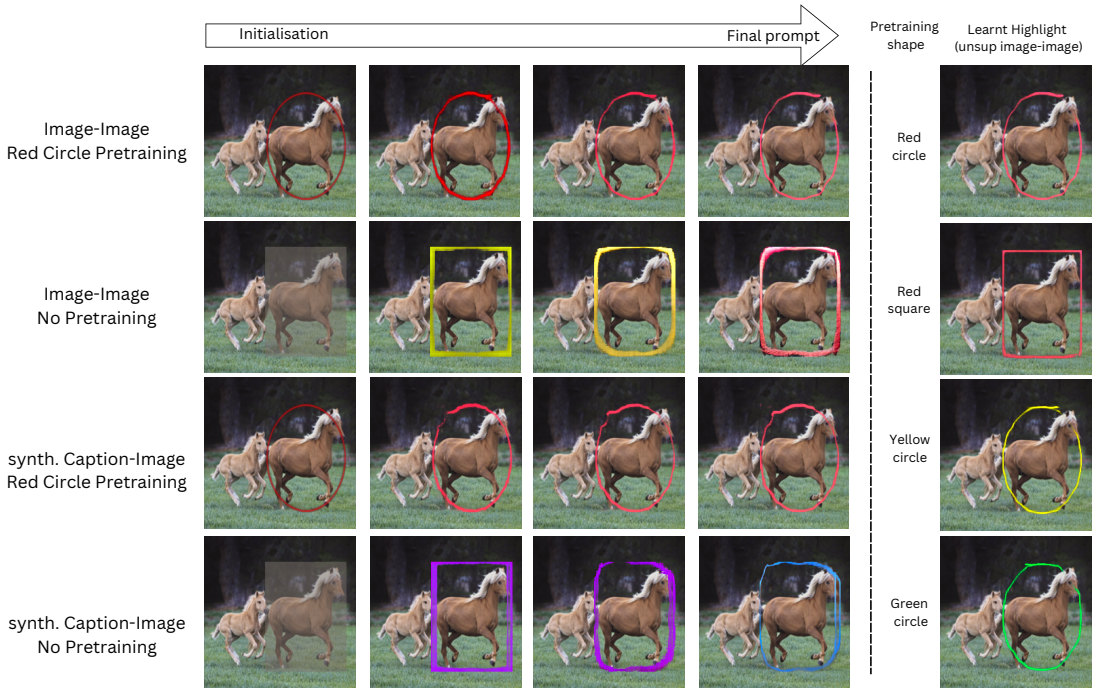


Figure 5.3: **Learnt visual markers.** Left: we show how the learnt visual markers in different training regimes change during training. We see that when no pretraining is used, Highlight converges to an outline of the prompted region, which is similar to the manually engineered prompts. Right: we show the final learnt prompts with several different pretraining visual markers.

Method	Pretraining	RefCOCO	RefCOCO+	RefCOCOg
Highlight (supervised)	✗	47.7	52.0	54.4
Highlight (supervised)	✓	45.2	49.5	50.7
Highlight (unsup, synth captions)	✗	41.2	46.2	47.2
Highlight (unsup, synth captions)	✓	41.9	47.0	48.2
Highlight (unsup, images only)	✗	34.3	39.9	41.9
Highlight (unsup, images only)	✓	42.5	47.6	48.4

Table 5.4: **Pretraining Highlight improves performance.** We find that pretraining Highlight to output a visual prompt (red circle) leads to better performance in the unsupervised regime, with the biggest gains seen for the images-only model.

better than single prompt methods but have a significant inference overhead. For this comparison we use the CLIP ViT-L/14@336 model, which is the most capable model used in the ensemble. Our supervised visual prompt outperforms the ensemble methods of ReCLIP, which uses post-processing with spatial relations and an ensemble of 2 models and 2 visual prompts for a total of 4 forward passes of the CLIP vision encoder, and performs on par with RedCircle, which uses an ensemble of 2 models and 3 visual prompts for a total of 6 forward passes of a CLIP vision encoder. In comparison, the Highlight prompt achieves similar performance, while only doing one pass of the CLIP vision encoder.

Unsupervised Highlight. In Table 5.1, we also see the performance of the unsupervised version of Highlight in two settings—using synthetic captions and images only. We outperform other visual prompts on all REC datasets. We see the Highlight prompt learnt with images only outperforms the one learnt with synthetic captions. Despite the modality gap [V. W. Liang et al. 2022] when training without text, our images-only framework is not affected by ambiguous synthetic captions, as discussed in Section 5.3.3.

Comparison to Guided Attention (GA) [Rezaei et al. 2024]. The concurrent work of [Rezaei et al. 2024] also learns a visual prompt from image data. Note, that GA is evaluated only using bounding boxes that are present in RefCOCO. Any sample that only has one possible bounding box is excluded. Thus, performances are higher than evaluating with proposals from MattNet. In Table 5.2 we observe, that our unsupervised Highlight learnt from images performs 18.2% better than GA. Contrary to our approach GA fixes the alpha mask $\mathcal{F}_\alpha$ to be a circle shape, for a fair comparison we thus also train a Highlight with a fixed alpha mask and only learn to colour in the mask. This Highlight then has as many degrees of freedom as GA, and still performs 18% better than GA. Thus, we conclude that Highlight's learning objective leads to a significantly better visual prompt than the one learnt by GA. Furthermore, whereas GA (i) requires the vision encoder to be a ViT and (ii) uses the inner representation of the vision encoder, Highlight (i) is architecture agnostic, and (ii) treats the vision encoder as a black box and only uses its output.

Effect of the target prompting function $\mathcal{F}^*$ We train Highlight with different target prompting functions $\mathcal{F}^*$ as part of the unsupervised image-image loss in Eq. (5.5). We see that learning Highlight using Blur (blurring everything outside the bounding box) leads to superior performance compared to training with a red circle prompt or with a crop. Also, using Crop as the target prompting function $\mathcal{F}^*$ leads to the biggest absolute gains over $\mathcal{F}^*$. We attribute this to the fact that we pretrain Highlight to output a red circle, which can then capture a complimentary signal from the crop/blur target functions. Still, we observe that the performance of the visual prompt learnt when the target $\mathcal{F}^*$ is a red circle outperforms using a plain red circle as the visual prompt.

Table 5.5: **Pretraining Highlight to output different shapes.** Here we show results for our unsupervised images-only model. We find that pretraining using a red circle leads to the best results. All pretrainings outperform a Highlight that was not pretrained at all.

Pretrain Shape	RefCOCO			RefCOCO+			RefCOCOg	
	Val (🧍, 🪑)	TestA (🧍)	TestB (🪑)	Val (🧍, 🪑)	TestA (🧍)	TestB (🪑)	Test (🧍, 🪑)	Val (🧍, 🪑)
Red Circle	**42.1**	**49.3**	36.1	**47.6**	**54.6**	**40.7**	**48.5**	**48.2**
Red Square	41.5	47.6	**36.7**	46.4	52.4	38.9	47.1	46.5
Green Circle	41.7	49.0	36.1	47.0	54.2	40.4	47.5	47.5
Yellow Circle	39.8	46.4	34.5	45.0	51.6	38.6	47.0	46.0
No Pretrain	33.6	38.3	31.0	39.8	43.9	36.0	42.2	41.5

The importance of pretraining Table 5.4 shows that pretraining Highlight to output a manually engineered visual prompt (red circle in this case) improves results in the unsupervised regime. As we decrease the level of supervision, the learning signal becomes weaker, and pretraining becomes more crucial, *i.e.* pretraining significantly improves performance when training only with images but has limited effects when training fully supervised. However, we notice that pretraining leads Highlight to a local minimum as seen on the right of Fig. 5.3, where the final prompt has changed little compared to the initialisation. In Table 5.5, we show the performance of our image-only framework when pretraining with different visual prompts. We observe that Highlight, which is pretrained to output a red circle, outperforms the other pretraining shapes and colours. Moreover, we notice that both shape *and* colour of the pretraining prompt are important since changing the colour or changing the shape, *i.e.* from circle to square, leads to worse performance. Furthermore, we also notice that networks pretrained to produce better prompts tend to deliver better performance later on. For example, a red circle prompt performs better than a green circle prompt, which outperforms a yellow circle prompt [Shtedritski et al. 2023b]. This relationship remains consistent even after optimising each Highlight.

Qualitative results. In Fig. 5.3, we show how the visual prompts change during training, with different objectives and different pretraining. We see that the visual prompts that have been pretrained change little, and the subsequent training with our framework just refines them. On the right, we see this also holds when pretraining using other shapes and colours. The visual prompts learnt from scratch, however, resemble a contour around the object boundaries, which is similar to the manually engineered visual markers. This suggests that these emergent

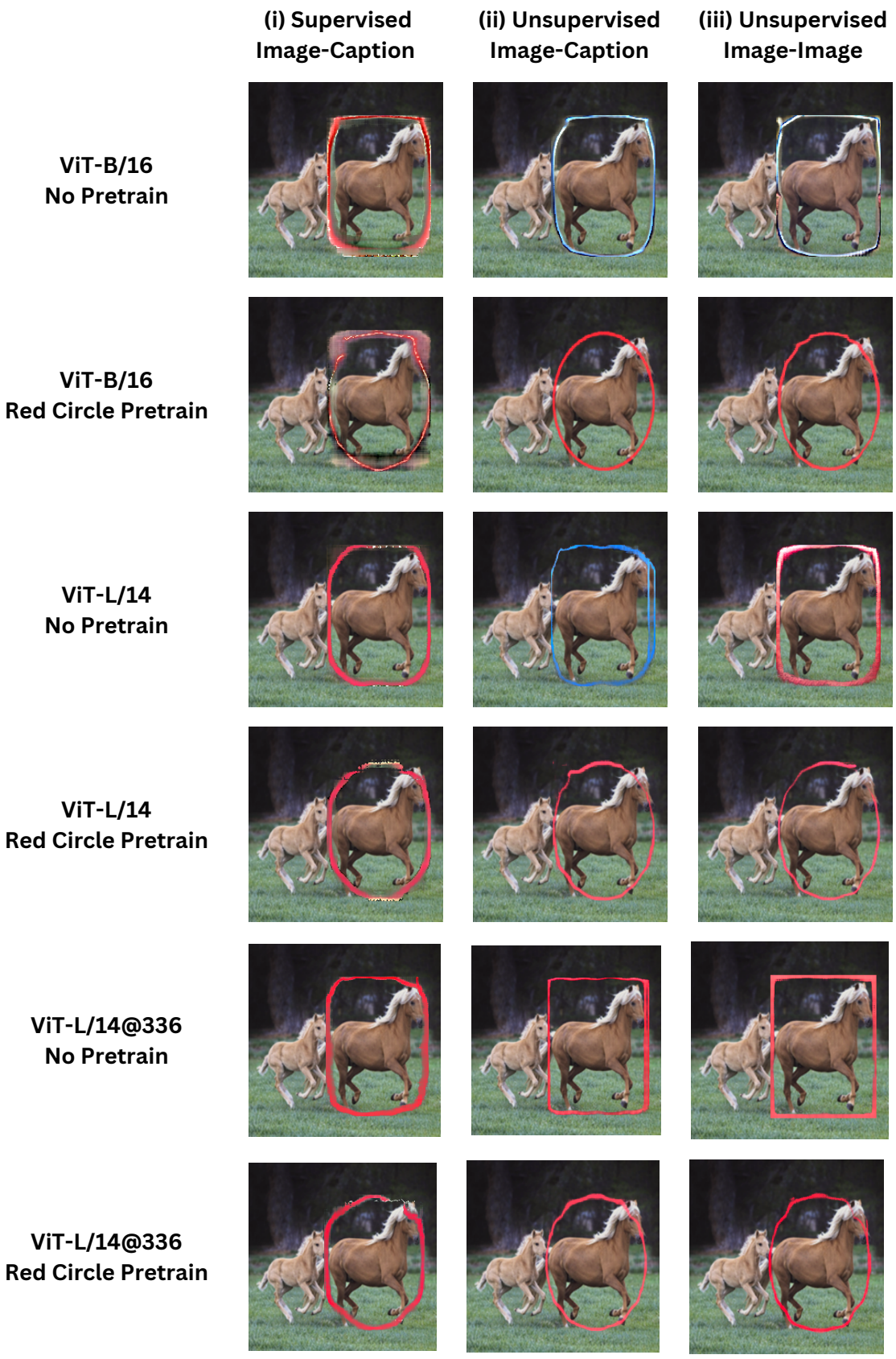


Figure 5.4: **Learnt visual markers.** We show the learnt visual marker for different CLIP models in the different supervision regimes.

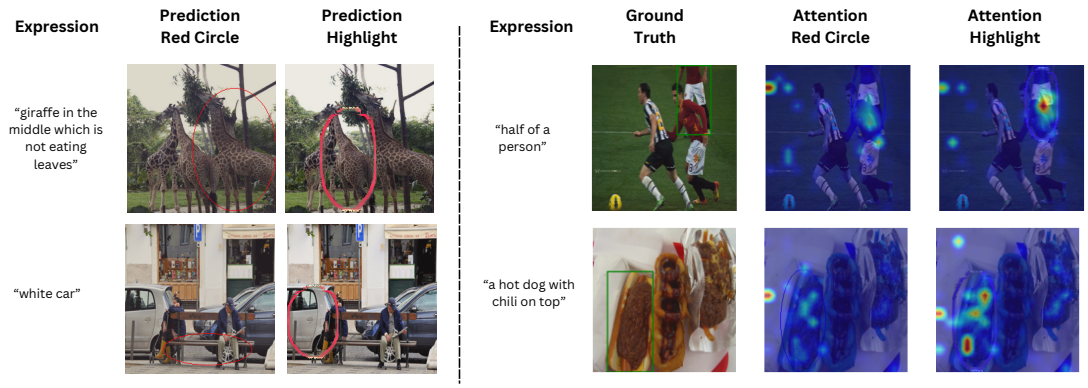


Figure 5.5: **Qualitative examples.** Left: We show examples of cases when supervised Highlight leads to a correct prediction on RefCOCO, but using a plain red circle fails at localising the correct region. Right: We show attention maps using the explainability method of [Chefer et al. 2021]. We see that when prompting with Highlight, we can successfully draw the attention of the visual transformer to the highlighted region.

behaviours of VLMs, where they can be prompted with visual markers, are indeed learnt from data. In Fig. 5.4 we show visual prompts learnt with different CLIP models, with and without pertaining. We see that this holds for the majority of models and supervision frameworks. For smaller models, such as the ViT-B/16, we see that the highlights are blurrier and less defined, as perhaps such models do not have the capacity necessary to learn such prompts from data, as suggested in [Shtedritski et al. 2023b], whose observation was purely based on performance results. In Fig. 5.5, we show attention maps generated using the method of [Chefer et al. 2021]. We see that Highlight can successfully steer the attention of the ViT to the highlighted object. Fig. 5.5 also shows examples where the predictions using a plain red circle fail, whereas Highlight correctly predicts the referred region.

5.5 Conclusion

Manually engineered visual prompts for VLMs have been extensively used for multiple tasks. However, choosing visual prompts requires trial and error and guessing. The framework we present, *Highlight*, automatically learns a visual prompt given a dataset of text and images, or images only. The optimised prompt can refine a manually engineered prompt, or be learnt from scratch, in which case we find they resemble manually engineered prompts. The visual prompts we automatically learn outperform manually engineered visual prompts, as well as computationally ensembles of models and prompts. We hope this inspires further work into auto-

matically learning to highlight images, as well as applying such methods beyond CLIP models and Referring Expression Comprehension.

Acknowledgements A. Shtedritski is supported by EPSRC EP/S024050/1. J. Zeller is supported by a fellowship of the German Academic Exchange Service (DAAD) and a scholarship from Google Deepmind.

5.6 Appendix

5.6.1 Adding Heuristical Spatial Relations

As detailed in Section 6.3, one potential caveat of training with synthetic captions is that captions might be ambiguous, *e.g.* in Fig. 5.2 the caption model might output the same caption "a horse" for both object proposals in the image. We experiment with alleviating this ambiguity by introducing spatial relations using the heuristics introduced by [Subramanian et al. 2022]: For each object proposal, we identify the closest other object proposal and then determine which spatial relations hold between the two bounding boxes (*e.g.* left, right, above, below, smaller, bigger) and select one of these at random. Next, we combine the captions from the two objects with a randomly chosen synonym for the selected spatial relation. We take the list of synonyms from [Subramanian et al. 2022]. For example, in Fig. 5.2, this process might change the captions from {"a horse", "a horse"} to {"a horse to the left of a horse", "a horse bigger than a horse"}.

Table 5.6 shows the performance of training with synthetic captions both with and without heuristically adding spatial relations. Overall, performance when training with heuristically added spatial relations is slightly better than training without spatial relations. Interestingly, we observe that this improvement mainly stems from TestA splits. This might indicate that the captions the caption model generates are better suited to differentiate objects from one another than humans.

Table 5.6: **Results for heuristically adding spatial relations.** Adding spatial relations to the unsupervised captions improves performance—mainly on TestA splits.

Method	RefCOCO			RefCOCO+			RefCOCOg	
	Val (🧍, 🐕)	TestA (🧍)	TestB (🐕)	Val (🧍, 🐕)	TestA (🧍)	TestB (🐕)	Test (🧍, 🐕)	Val (🧍, 🐕)
Highlight (unsup, synth. captions)	41.4	48.1	36.1	47.0	52.8	41.0	48.2	48.2
Highlight (unsup, synth. captions) + SR	41.4	48.7	36.1	47.0	53.6	40.8	48.3	49.2

Table 5.7: **Comparison to FGVP.** We compare to FGVP [Lingfeng Yang et al. 2024], which is a visual prompting method, which is supervised in that it requires masks from SAM [Kirillov et al. 2023b]. *SR* is the use of a spatial relations post-processing, that takes into account words like "left", "right", "bigger", "smaller", etc, which [Subramanian et al. 2022] found to boost performance. *SS* is the use of Score Subtraction, which [Shtedritski et al. 2023b] found to boost performance. **Num Enc** is the number of forward passes of the visual encoder (number of prompts used times the number of visual encoders ensembled) 🧍 contains expressions referring to people, 🪑 for objects. **Sup** indicates whether a method was supervised or not. The best result is marked in **bold**, second best is underlined.

Method	Sup	Num. Enc.	RefCOCO			RefCOCO+			RefCOCOg		Avg
			Val (🧍, 🪑)	TestA (🧍)	TestB (🪑)	Val (🧍, 🪑)	TestA (🧍)	TestB (🪑)	Test (🧍, 🪑)	Val (🧍, 🪑)	
Highlight (Unsup., synth captions)	✗	1	41.4	48.1	36.1	47.0	52.8	41.0	48.2	48.2	45.3
Highlight (Unsup., images only)	✗	1	42.1	49.3	36.1	47.6	54.6	40.7	48.5	48.2	45.9
FGVP	✓	1	46.1	53.0	40.4	50.4	57.5	42.6	54.5	54.1	49.8
FGVP^{SR}	✓	4	52.0	55.9	48.8	53.3	60.4	46.7	62.1	61.9	55.1
FGVP^{SS}	✓	6	53.9	60.2	44.3	59.3	66.6	48.8	61.0	61.3	56.9
FGVP^{SR+SS}	✓	8	**59.6**	**65.0**	**52.0**	**60.0**	**66.8**	**49.7**	**63.3**	**63.4**	**60.0**
Highlight (Supervised)	✓	1	49.3	57.7	40.6	54.8	62.5	45.1	56.5	56.6	52.9

5.6.2 Transfer between models

In Fig. 5.6, we see the relative performance improvement of Highlight over a network pretrained to output a red circle when trained on one CLIP model and evaluated on another. We see the biggest improvements when evaluating with the CLIP ViT-B/16 model and observe that the prompt optimised with CLIP ViT-L/14 is more transferable to other models than the other optimised prompts. Overall, a visual prompt learnt on a larger model transfers well on a smaller model, *e.g.* learning on ViT-L/14@336 and evaluating on ViT-B/16 work better than the opposite.

5.6.3 Comparison to FGVP

In Table 5.7 we compare Highlight to FGVP [Lingfeng Yang et al. 2024]. FGVP is supervised in that it uses SAM [Kirillov et al. 2023b] to generate mask outlines for objects, which are used for visual prompts. Additionally, FGVP utilises several post-processing techniques. In Table 5.7, we see that supervised Highlight outperforms FGVP when it uses a single prompt. Supervised highlight has a similar performance to FGVP when it uses up to 6 forward passes of the visual encoder and either of the two ad-hoc post-processing techniques. However, when FGVP uses 8 passes of the visual encoder and both post-processing methods, it performs better than Highlight. Overall, Highlight is a much more lightweight and efficient method, and performs favourably to FGVP.

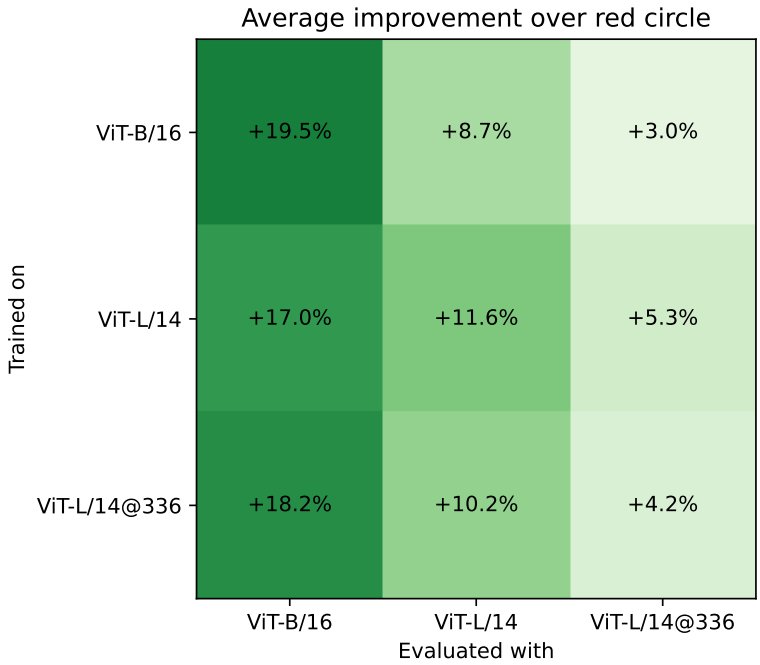


Figure 5.6: **Transferability of an optimised Highlight trained for one CLIP model to other models.** We use Highlights that were optimised with our image-image loss and pretrained to output a red circle for evaluation on other CLIP models. We observe that even though the Highlights were not optimised for that specific CLIP model, they still increase performance. We report the average performance increases across all RefCOCO test and validation datasets.

5.6.4 Implementation details

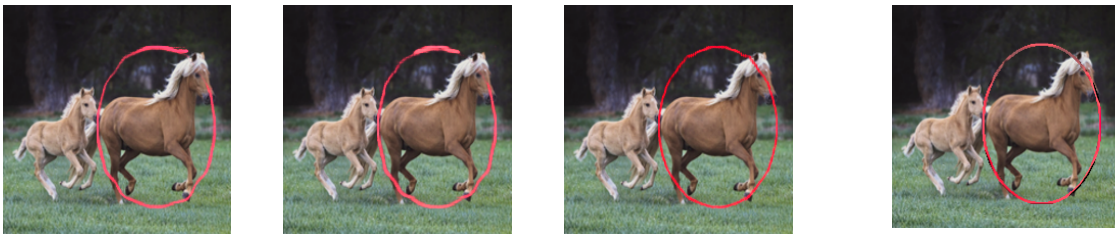
Table 5.9 gives an overview of the different hyperparameters for all runs. Note that if we train supervised, we iterate over referring expressions and corresponding bounding boxes instead of images. Therefore, we train for fewer but larger epochs. Moreover, we empirically observe that training in the supervised case leads to invisible prompts. Thus, we add a visibility loss as introduced in Eq. (5.6) with a threshold τ of 10%. For the pretrained $\text{Highlight}_{\text{im2im}}$ for the ViT-B/16 model, we use $\mathcal{F}^*$ crop instead of blur. To adjust to the higher resolution, we filter for object proposals with a minimum height and width of 75px for the ViT-L/14@336 model. Therefore, we consider the same object proposals for both resolutions. We adjust batch size and minibatch size in order to fit on a single GPU, depending on the CLIP model and supervision regime used. Minibatch size denotes how many samples are considered at maximum for one iteration of the contrastive loss, *e.g.* if the minibatch size is 10, but there are 15 valid object proposals, they are cut down to 10 object proposals (at random), and for each positive pair in the contrastive loss, there are 9 negatives.

Table 5.8: **Results for different CLIP models**. We train Highlight in the three different modes, (i) unsupervised im2im, using image-image pairs only, (ii) unsupervised t2im, using synthetic text-image pairs, and (iii) supervised t2im, using ground truth text-image pairs. Overall, we see that pretraining improves performance in the unsupervised image-image regime, and does not help or hurts performance when either ground-truth or unsupervised captions are used.

Model	Method	Supervision	Pretrain	RefCOCO			RefCOCO+			RefCOCOg	
				Val (🧍, 👓)	TestA (🧍)	TestB (👓)	Val (🧍, 👓)	TestA (🧍)	TestB (👓)	Test (🧍, 👓)	Val (🧍, 👓)
ViT-B/16	Crop	—	—	31.1	31.5	33.4	35.9	37.5	36.6	50.6	51.0
	Blur	—	—	35.5	39.5	32.0	39.0	45.1	35.1	47.5	47.3
	Red Circle	—	—	36.7	42.0	31.4	39.8	44.9	34.0	40.4	40.5
	Highlight$_{\text{im2im}}$	✗	✓	39.7	44.7	33.4	43.3	48.6	36.0	42.3	42.3
	Highlight$_{\text{im2im}}$	✗	✗	35.2	38.0	33.6	38.2	39.8	36.0	40.7	41.2
	Highlight$_{\text{t2im}}$	✗	✓	38.5	43.8	32.1	42.1	47.8	35.2	41.4	41.5
	Highlight$_{\text{t2im}}$	✗	✗	37.9	42.0	34.7	41.7	45.1	37.6	43.0	42.5
	Highlight$_{\text{t2im}}$	✓	✓	44.9	52.7	37.3	47.8	56.4	39.4	47.6	47.4
	Highlight$_{\text{t2im}}$	✓	✗	45.0	52.5	36.9	47.3	54.9	40.0	48.0	47.6
ViT-L/14	Crop	—	—	27.9	29.0	29.4	32.7	33.8	33.0	45.9	47.0
	Blur	—	—	32.2	38.1	29.3	36.7	42.5	31.1	42.3	42.3
	Red Circle	—	—	39.8	47.1	33.2	44.5	51.5	38.2	48.4	48.0
	Highlight$_{\text{im2im}}$	✗	✓	42.1	49.3	36.1	47.6	54.6	40.7	48.5	48.2
	Highlight$_{\text{im2im}}$	✗	✗	33.6	38.3	31.0	39.8	43.9	36.0	42.2	41.5
	Highlight$_{\text{t2im}}$	✗	✓	41.4	48.1	36.1	47.0	52.8	41.0	48.2	48.2
	Highlight$_{\text{t2im}}$	✗	✗	40.2	46.9	36.5	46.1	51.3	41.0	47.4	47.0
	Highlight$_{\text{t2im}}$	✓	✓	45.0	52.3	38.4	49.6	56.5	42.5	51.1	50.3
	Highlight$_{\text{t2im}}$	✓	✗	47.6	56.1	39.4	52.7	60.5	42.9	55.1	53.7
MetaCLIP ViT-L/14 (400m)	Crop	—	—	31.0	32.8	31.8	36.0	36.6	37.3	47.5	47.8
	Blur	—	—	24.8	25.5	24.6	28.3	28.1	27.5	35.9	35.9
	Red Circle	—	—	34.8	38.8	32.6	37.7	41.1	36.1	43.0	42.0
	Highlight$_{\text{im2im}}$	✗	✓	36.9	41.0	34.0	39.9	42.8	37.6	43.3	43.3
	Highlight$_{\text{im2im}}$	✗	✗	29.3	30.5	30.2	32.0	32.3	32.3	38.6	39.2
	Highlight$_{\text{t2im}}$	✓	✓	39.5	44.6	35.3	43.3	46.6	38.5	46.0	46.5
	Highlight$_{\text{t2im}}$	✓	✗	37.5	41.4	35.3	40.8	44.8	39.2	45.6	45.6
MetaCLIP ViT-L/14 (2.5B)	Crop	—	—	37.7	40.7	35.8	40.6	42.6	39.5	50.5	50.1
	Blur	—	—	28.9	32.1	27.1	31.2	34.8	29.0	38.9	39.3
	Red Circle	—	—	38.0	44.5	31.2	41.4	47.0	35.4	43.7	43.2
	Highlight$_{\text{im2im}}$	✗	✓	40.2	46.3	34.2	44.2	49.2	38.6	46.4	46.8
	Highlight$_{\text{im2im}}$	✗	✗	31.0	29.5	31.1	32.1	30.9	33.5	37.2	37.7
	Highlight$_{\text{t2im}}$	✓	✓	43.5	51.5	37.5	47.7	53.2	40.2	48.8	47.8
	Highlight$_{\text{t2im}}$	✓	✗	42.6	47.4	37.0	46.0	49.2	41.2	48.2	47.9
ViT-L/14@336	Crop	—	—	28.7	29.6	29.9	33.7	33.9	33.6	46.9	48.0
	Blur	—	—	31.3	36.1	28.7	36.2	40.9	30.8	42.4	42.1
	Red Circle	—	—	41.2	50.4	34.3	46.2	54.3	39.3	49.9	49.3
	Highlight$_{\text{im2im}}$	✗	✓	41.2	49.0	34.5	46.2	54.3	38.9	48.4	48.0
	Highlight$_{\text{im2im}}$	✗	✗	38.5	44.1	33.4	42.9	47.4	37.8	43.2	43.5
	Highlight$_{\text{t2im}}$	✗	✓	42.1	50.5	34.9	46.8	54.5	39.7	48.3	48.7
	Highlight$_{\text{t2im}}$	✗	✗	41.2	47.7	35.9	46.6	52.5	39.6	48.4	48.8
	Highlight$_{\text{t2im}}$	✓	✓	47.3	57.1	37.7	52.3	61.5	42.9	53.1	52.8
	Highlight$_{\text{t2im}}$	✓	✗	49.3	57.7	40.6	54.8	62.5	45.1	56.5	56.6

5.6.5 Additional qualitative results

In Fig. 5.4, we show visual prompts we learn with Highlight for different CLIP models in all the supervision regimes we consider. We see that all models learn a contour in the unsupervised regime, where for the large models, this contour is red—resembling a red circle. In Fig. 5.7, we show (i) visual markers optimised for different target functions when using the unsupervised image-image objective and red circle pretraining and (ii) the learn visual marker when we compare to [Rezaei et al. 2024], fixing the prompt shape to a circle.



Highlights optimized for different $\mathcal{F}^*$. **Fixed mask.**

Figure 5.7: **Learnt visual prompts for ViT-L/14 trained with unsup. image-image loss.** Left: We show the learnt markers when optimising a Highlight pretrained with a red circle using different target functions. Right: We show the learnt prompt when training a Highlight with a restricted prompt shape to a circle. Similarly to [Rezaei et al. 2024], we parametrise the prompt network with a UNet for this experiment.

Table 5.9: **Hyperparameters for all optimised Highlights.**

Model	Method	Sup	Pretrain	Learning Rate	β	γ	Scheduler	Training Samples (Epochs)	Batch Size	Minibatch Size	Temperature
ViT-B/16	$\text{Highlight}_{\text{im2im}}$	✗	✓	0.001	0.0001	0	✓	∼100,000 (4)	20	10	20
	$\text{Highlight}_{\text{im2im}}$	✗	✗	0.001	0.0001	0	✗	∼200,000 (8)	56	15	20
	$\text{Highlight}_{\text{t2im}}$	✗	✓	0.001	0.0001	0	✓	∼100,000 (4)	20	10	20
	$\text{Highlight}_{\text{t2im}}$	✗	✗	0.001	0.0001	0	✗	∼200,000 (8)	20	10	20
	$\text{Highlight}_{\text{t2im}}$	✓	✓	0.001	0.5	1.5	✓	∼150,000 (1)	16	25	20
	$\text{Highlight}_{\text{t2im}}$	✓	✗	0.001	0.05	0.15	✗	∼300,000 (2)	16	25	20
ViT-L/14	$\text{Highlight}_{\text{im2im}}$	✗	✓	0.001	0.0001	0	✓	∼100,000 (4)	20	10	20
	$\text{Highlight}_{\text{im2im}}$	✗	✗	0.001	0.0001	0	✗	∼200,000 (8)	20	10	20
	$\text{Highlight}_{\text{t2im}}$	✗	✓	0.001	0.001	0	✓	∼100,000 (4)	20	10	20
	$\text{Highlight}_{\text{t2im}}$	✗	✗	0.001	0.0001	0	✗	∼200,000 (8)	20	10	20
	$\text{Highlight}_{\text{t2im}}$	✓	✓	0.001	0.5	1.5	✓	∼150,000 (1)	16	14	20
	$\text{Highlight}_{\text{t2im}}$	✓	✗	0.001	0.05	0.15	✗	∼600,000 (4)	16	14	20
ViT-L/14@336	$\text{Highlight}_{\text{im2im}}$	✗	✓	0.001	0.00001	0	✓	∼100,000 (4)	20	10	20
	$\text{Highlight}_{\text{im2im}}$	✗	✗	0.001	0.00001	0	✗	∼25,000 (1)	20	10	20
	$\text{Highlight}_{\text{t2im}}$	✗	✓	0.001	0.0001	0	✓	∼100,000 (4)	20	10	20
	$\text{Highlight}_{\text{t2im}}$	✗	✗	0.001	0.00001	0	✗	∼100,000 (4)	20	10	20
	$\text{Highlight}_{\text{t2im}}$	✓	✓	0.001	0.1	0.3	✓	∼150,000 (1)	16	19	20
	$\text{Highlight}_{\text{t2im}}$	✓	✗	0.001	0.05	0.15	✗	∼450,000 (3)	16	19	20

Chapter 6

SynCity: Training-Free Generation of 3D Worlds

This paper will be presented at ICCV 2025.

SynCity: Training-Free Generation of 3D Worlds

Paul Engslter* Aleksandar Shtedritski*

Iro laina Christian Rupprecht Andrea Vedaldi

Visual Geometry Group, University of Oxford

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

September 29, 2025



Figure 6.1: We introduce **SynCity**, a novel method that can generate from a prompt complex and immersive 3D worlds that can be navigated freely. Our method is training-free and leverages powerful language, 2D and 3D generators via novel prompt engineering strategies.

Abstract

We address the challenge of generating 3D worlds from textual descriptions. We propose SynCity, a training- and optimization-free approach, which leverages the geometric precision of pre-trained 3D generative models and the artistic versatility of 2D image generators to create large, high-quality 3D spaces. While most 3D generative models are object-centric and cannot generate large-scale worlds, we show how 3D and 2D generators can be combined to generate ever-expanding scenes. Through a tile-based approach, we allow fine-grained control over the lay-

out and the appearance of scenes. The world is generated tile-by-tile, and each new tile is generated within its world-context and then fused with the scene. SynCity generates compelling and immersive scenes that are rich in detail and diversity.

6.1 Introduction

We consider the problem of generating 3D worlds from textual descriptions. Generating 3D content, for example, for video games, virtual reality, special effects and simulation, is highly laborious and time-consuming. When it comes to generating entire 3D scenes, much of the content is not of particular artistic value, and its manual creation, which is still necessary, may be seen as a waste of human resources, talent and creativity. Generative models can help to reduce or even remove this burden by largely automating many of these mundane tasks.

The advent of modern generative AI has significantly impacted 3D content generation and promises to reduce the cost of its production dramatically. DreamFusion [Poole et al. 2023] was among the first to co-opt state-of-the-art diffusion-based 2D image generators [Rombach et al. 2022] to create 3D objects. The area has since matured significantly by fine-tuning 2D image generators to produce multiple consistent views of an object [Yichun Shi et al. 2024; Szymanowicz et al. 2023; Melas-Kyriazi et al. 2024; R. Gao et al. 2024] and by learning few-view 3D reconstruction networks [Jiahao Li et al. 2024; Y. Xu et al. 2024]. More recently, the focus has shifted to methods that learn a 3D latent space [B. Zhang et al. 2023; Deemos 2024; Yangguang Li et al. 2025; Weiyu Li et al. 2024; Xiang et al. 2024a], which can then be sampled to generate 3D objects. Because the latent space directly encodes the 3D structure of the object rather than its 2D appearance, these methods can generate much more accurate and regular geometry.

Despite their advantages, 3D generative methods have been so far mostly limited to generating individual objects. However, the most promising usage of 3D generative AI is the construction of entire virtual worlds, as this is where automation can make by far the most difference. There is ample literature on generating 3D scenes from textual or image prompts. Most such methods are image-based, and progressively reconstruct larger scene regions by expanding from an initial image [Rockwell et al. 2021; Höllein et al. 2023; Chung et al. 2023; H.-X. Yu et al. 2024b; Jingbo Zhang

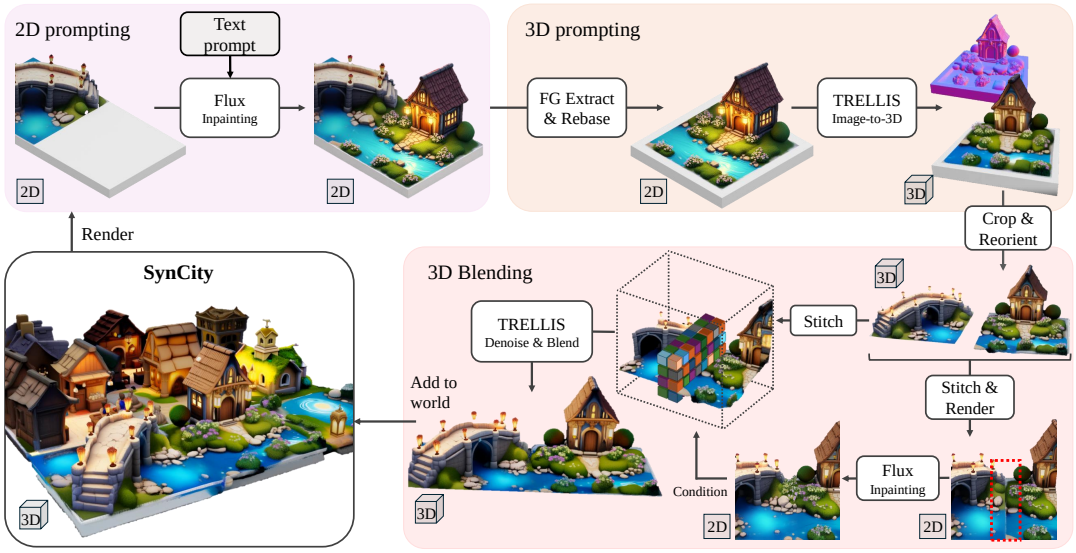


Figure 6.2: **Overview of SynCity.** 2D prompting: To generate a new tile, we first render a view of where that tile should be placed, including context from neighbouring tiles. 3D prompting: We extract the new tile image and construct an image prompt for TRELLIS by adding a wider base under the tile. 3D blending: The 3D model that TRELLIS outputs is usually not well blended with the rest of the scene. To address that, we render a view of the new tile next to each neighbouring tile, and inpaint the region between the two with an image inpainting model. Next, we condition using that well-blended view to refine the region between the two 3D tiles. Finally, the new, blended, tile is added to the world.

et al. 2024; A. Liu et al. 2021; Fridman et al. 2023; Ouyang et al. 2023; Engstler et al. 2025b], combining depth prediction, image and depth outpainting, and 3D reconstruction using NeRF [Mildenhall et al. 2020] or 3D Gaussian Splatting [Kerbl et al. 2023]. The main advantage of these approaches is that they can leverage powerful 2D image generator models to create the first and subsequent views of the scene. These 2D generators allow the overall system to understand complex textual prompts and generate corresponding 3D scenes with good artistic quality. However, it is difficult for these approaches to maintain a coherent 3D structure across a large scene. For example, while the reconstructed scene may envelop the observer in a 360° manner, it is generally not possible to 'walk' into the scene for more than a few steps. This is the case even for state-of-the-art implementations like the one of World Labs [World Labs 2024], a company specialized in SpatialAI.

A challenge with extending scenes beyond these '3D bubbles' is that it is difficult for image-based methods to maintain consistency incrementally without drifting. We argue that 3D generative models might be preferable as they can regularize and constrain the reconstructed geometry, including hallucinating shape and textures

in regions *behind* the visible sides of objects. This is clearly shown in prior works like BlockDiffusion [Z. Wu et al. 2024] and LT3SD [Meng et al. 2024], where large coherent spaces can be generated. However, these methods are limited in the quality and diversity of the generated scenes as it is difficult to train 3D generative models directly for scene generation. In particular, unlike their view-based peers, these methods do not build on an image generator. They thus cannot benefit from the artistic quality and ability to interpret complex textual prompts that come from pre-training 2D generators on billions of images.

In this work, we thus seek to build on 3D generative models while still building on the strengths of 2D image generators to generate large, high-quality 3D spaces that can be navigated freely (Fig. 6.1). First, we note that while 3D models like TRELLIS [Xiang et al. 2024a] are trained for object-level reconstruction, they can reconstruct fairly complex local compositions of multiple objects. Borrowing ideas from video game world building, we show in particular that TRELLIS can effectively generate, if not an entire world, at least a *tile* representing a local region of the world. We show, in particular, how to prompt the model with an 'isometric' view of the tile and then generate the tile in 3D.

Given this basic capability, we then look at the problem of generating a large scene by generating and stitching together multiple tiles. We build on a text-to-image generator (Flux [Labs 2023]) and propose a novel way of prompting that *stabilizes* it to consistently produce tiles with a similar isometric framing. In this manner, we encourage the tiles' framing to be stable and compatible between different tiles, making it easier to stitch them together.

To ensure tiles to fit together appropriately, we propose two mechanisms. First, we encourage consistency in appearance by using previously generated tiles to draw context for the image generator, where each new tile *inpaints* a missing region in a 2D isometric view of the scene. Secondly, we enforce geometric consistency by blending the 3D representations of neighbouring tiles using the 3D generative model.

6.2 Related Work

Novel view synthesis for scenes. Expanding an image beyond its boundaries has been a long-standing task in computer vision. Early methods that sought to expand object-centric scenes rely on layer-structured representations [Tulsiani et al. 2018; Mildenhall et al. 2019; Srinivasan et al. 2019; Tucker and Snavely 2020; Jiaxin Li et al. 2021; Shih et al. 2020], which disregard the scene's true geometry. SynSin [Wiles et al. 2020] is a pivotal work, where image features are projected and used as conditioning to generate novel views, achieving geometric and semantic consistency. ZeroNVS [Sargent et al. 2023] introduces high-quality results with fine-grained control of the camera but remains object-centric. GenWarp [J. Seo et al. 2024] integrates semantic information through cross-attention when generating a novel view.

The major challenges for these methods remain semantic drift and object permanence. To obtain an explicit 3D representation, the generated views need to be transferred into such a representation, e.g., NeRF [Mildenhall et al. 2020] or Gaussians [Kerbl et al. 2023; B. Huang et al. 2024], where any geometric conflicts would need to be resolved.

Image projection-based scene generation. A different line of work follows the paradigm of building the 3D representation of a scene sequentially using 2D image generation models [Rockwell et al. 2021; Zhengyi Wang et al. 2023; Höllein et al. 2023; Fridman et al. 2023; Lei et al. 2023; Xiang et al. 2023; Chung et al. 2023; H.-X. Yu et al. 2024b; Jingbo Zhang et al. 2024; Ouyang et al. 2023]. Most of them employ an image generation model to outpaint the existing scene using pre-defined camera poses. The results are then fused in 3D with depth prediction models. Text2Room [Höllein et al. 2023] generates meshes of indoor scenes. As the scene is clearly delimited by the bounds of the mesh, it can be freely explored. LucidDreamer [Chung et al. 2023] and Text2Immersion [Ouyang et al. 2023] go beyond indoor scenes, but their generated scenes reveal geometric inconsistencies when stepping away from the camera poses used to generate the scene. Invisible Stitch [Engstler et al. 2025b] addresses this issue by inpainting depth (rather than naively aligning it) and RealmDreamer [Shriram et al. 2025] proposes multiple optimization losses to refine the generated scene. Despite these

improvements, the resulting scenes still suffer from geometric artifacts and remain rather small. WonderJourney [H.-X. Yu et al. 2024b] introduces novel ideas for depth fusion, such as grouping objects at similar disparity to planes and sky depth refinement, enabling large 'scene journeys', where independent representations are built between scene 'keyframes', but these are not merged into one coherent scene. WonderWorld [H.-X. Yu et al. 2024a] leverages these improvements to build a single scene, allowing interactive updates, but the true extent of the generated scenes remains limited. Other works use panoramas [G. Wang et al. 2024; Stan et al. 2023] or implicit representations [Bautista et al. 2022; Sargent et al. 2023] but the freedom of movement remains constricted.

Procedural scene generation. Further methods permit long-range fly-overs over nature [A. Liu et al. 2021; Z. Li et al. 2022; S. Cai et al. 2023; Zhaoxi Chen et al. 2023; Chai et al. 2023] or cities [C. H. Lin et al. 2023; H. Xie et al. 2024; Y. Shang et al. 2024]. These usually generate procedural unbounded images (e.g., the terrain make-up or a city layout). While those methods create realistic-looking images, they are often monotonous as the methods are domain-specific and thus highly constrained in the variety they can generate.

3D scene generation. Instead of merely generating images of a scene or outpainting it only in 2D, further methods generate the representation directly. Set-the-Scene [Cohen-Bar et al. 2023] adds a layer of control to the layout of NeRF scenes by defining object proxies. BlockFusion [Z. Wu et al. 2024] learns a network to auto-regressively diffuse small blocks to extend a mesh. A 2D layout conditioning is used to control the generation process, allowing users to generate scenes of rooms, a village, and a city. While the method allows building large-scale scenes, the variety of the objects it generates is severely limited as it requires domain-specific 3D training data. Furthermore, it generates untextured meshes. LT3SD [Meng et al. 2024] learns a diffusion model that generates 3D environments in a patch-by-patch and coarse-to-fine fashion. However, this method is only trained to produce indoor scenes.

At the same time, the synthesis of complex, high-fidelity objects has been enabled by the rapid progress in the fields of text-to-3D and image-to-3D generation [C. Lin et al. 2022; R. Liu et al. 2023a; Qian et al. 2023; Yichun Shi et al. 2024; Raj

et al. 2023; J. Tang et al. 2024; Yangguang Li et al. 2025; Longwen Zhang et al. 2024; Weiyu Li et al. 2024; Xiang et al. 2024a; Z. Zhao et al. 2025]. Trained on large-scale curated subsets of 3D datasets such as Objaverse-XL [Deitke et al. 2023b], these models can generate a large variety of different objects.

However, to the best of our knowledge, no prior work has leveraged object generators for scene generation.

6.3 Method

Our goal is to generate a 3D world $\mathcal{G}$ from an initial textual prompt p_0. Our main result is to show how prompt engineering can be used in combination with *off-the-shelf* language, 2D and 3D generators to create the entire world automatically, with no need to retrain the models.

We structure the world as a $W \times H$ grid $\mathcal{T} = \{0, \ldots, W-1\} \times \{0, \ldots, H-1\}$ of square tiles, each of which can contain several complex 3D objects (e.g., a building, a bridge, trees, etc.) as well as the ground surface.We generate the world progressively, tile by tile, as shown in in Fig. 6.3. Hence, when tile $(x, y) \in \mathcal{T}$ is generated, tiles $\mathcal{T}(x, y) = \{(x', y') \in \mathcal{T} : y' < y \vee (y' = y \wedge x' < x)\}$ have already been generated.

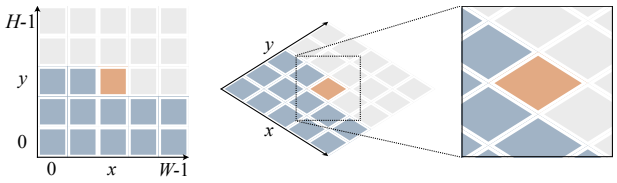


Figure 6.3: *Left:* Progressive generation of world tiles $\mathcal{T}$. *Right:* Isometric framing of a tile for image-based prompting.

An overview of our approach is shown in Fig. 6.2. The first step of our method is to expand the world description p_0 into tile-specific prompts (Section 6.3.1). The second step is to pass these tile-specific prompts to a 2D image generator and inpainter to create an isometric view of each tile, accounting for the part of the world generated so far (Section 6.3.2). The third step is to extract image prompts from these isometric views and use them as input to a text-to-3D generator to reconstruct each tile's geometry and appearance in 3D (Section 6.3.3). The final step is to align and blend the 3D reconstructions of the tiles to create a coherent

3D world (Section 6.3.4).

6.3.1 Prompting the Language Model

The goal of language prompting is to take a high-level textual description of the world p_0 and expand it into a set p of tile-specific textual prompts that can be used to generate the 3D world. Specifically, p is a collection of sub-prompts $p_{xy} \in \Sigma^*$, one for each tile, and a world-level 'style' prompt $p_\star \in \Sigma^*$, so that we can write $p = \{p_{xy}\}_{(x,y)\in\mathcal{T}} \cup \{p_\star\}$, where Σ^* is the set of all possible strings.

The prompt p could be constructed manually (which allows controlling the content of each tile) or generated by a large language model (LLM) such as ChatGPT [OpenAI 2023] from a 'seed' prompt. For the latter, we prompt ChatGPT o3-mini-high to generate a grid-like world with tile-specific descriptions, providing it with an example JSON file (see Section 6.6.1 for details).

6.3.2 Prompting the 2D Generator

We use the language prompts p from Section 6.3.1 to prompt an off-the-shelf 2D image generator Φ_{2D} to output a 2D image $I(x, y)$ of each tile to be generated, as shown for example in Fig. 6.4. The image $I(x, y)$ must satisfy several constraints: (1) It must reflect the tile-specific instructions p_{xy} of the target tile as well as the world-level instructions $p_\star$. (2) It must be suitable for prompting the image-to-3D generator in the next step. (3) It must be consistent with the previously generated tiles.

Our prompting strategy is designed to satisfy these constraints. The image is drawn as a sample $I(x, y) \sim \Phi_{\text{2D}}(q, B, M)$ from the 2D image generator Φ_{2D}, where $q = p_{xy} \cdot p_\star$ is a prompt that combines the tile-specific and world-level descriptions. The generator Φ_{2D} is also provided with a base image B and an inpainting mask M that constrain the output. We assume that Φ_{2D} is capable of inpainting — a common feature of modern image generators.

Tile inpainting. To satisfy constraint (2), we need to encourage the image generator to generate regular tiles so that the image-to-3D model can output tiles with regular geometry that fit well together. We assume that tiles have a fixed square basis of unit size and that they are imaged in an 'isometric' manner. This

framing of the tiles is conducive to the generation of regular 3D tiles. Furthermore, it is a common choice in video games and might have been observed by the image generator during training as these are often trained on game-like data.

Hence, our goal is to condition the image generator Φ_{2D} to produce images of this kind. While one possible approach is to fine-tune the generator on such images, we demonstrate that it is possible to obtain this effect through prompt engineering alone, avoiding any retraining. We achieve this by carefully constructing the inputs B and M as shown in Fig. 6.4. Specifically, we set B to be the image of the base of the tile, as a square, grey slab imaged from a fixed isometric vantage point. The mask M is a binary mask covering a cube on top of the base.

Figure 6.4 shows the result of prompting the model in this manner as well as what happens if signals B and M are removed: the viewpoint and general frame of the tile is random and not suitable for 3D generation.

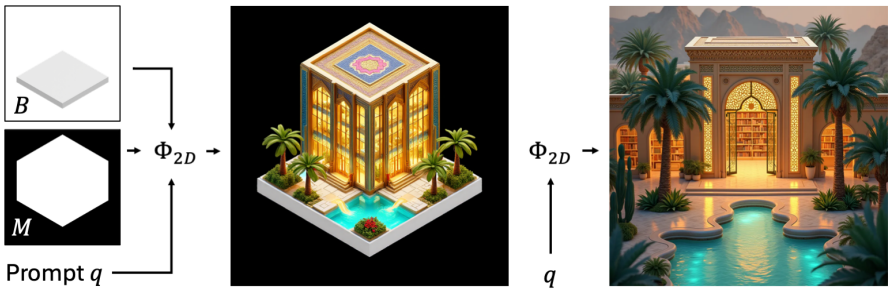


Figure 6.4: *Left:* Generation of the 2D image prompt for the first world tile at $x = 0$ and $y = 0$. The image generator Φ_{2D} is conditioned on $q = p_{00} \cdot p_\star$ and tasked with inpainting the base image B in the masked region M. *Right:* If we do not 'frame' the image by using B and M, the generator produces an image which is not suitable for tiling.

Taking the context into account. Except for the first tile $(0, 0)$, the tiles are generated in the context of the world already generated. In order to account for this context, for tiles with $x, y > 0$, we modify the base image B and the mask M as shown in Fig. 6.5. For the base image B, we render the part of the 3D world generated so far, providing context for the inpainting network. We also modify the mask M to avoid covering tiles already generated to the left (west, i.e. for a tile $(i, j) \in \mathcal{T}$ these are tiles $\{(x, y) \in \mathcal{T} : x < i \wedge y = j\}$).

Because we wish to ensure continuity of the ground, before rendering this contextual image, we trim any 3D geometry that is sufficiently high to occlude the tile we wish to generate, as shown in Fig. 6.6 (see the result in Fig. 6.5).



Figure 6.5: *Left:* Base image B and inpainting mask M (white overlay) to prompt the image generator Φ_{2D} to generate an image for a $x > 0$, $y > 0$ world tile. *Right:* Result of inpainting.

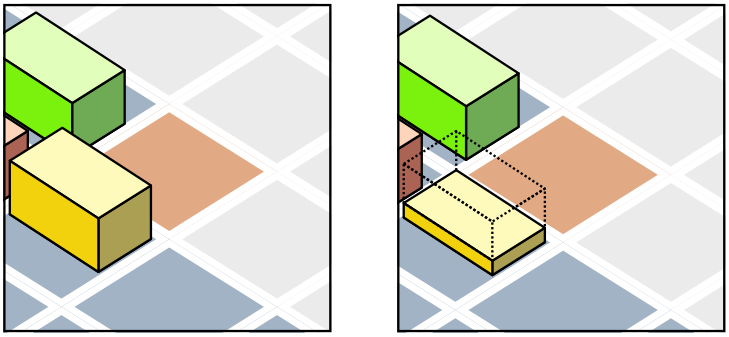


Figure 6.6: Trimming tall structures for 2D prompting.

The appendix discusses a special case for tiles at the boundaries of the world (see Section 6.6.2).

6.3.3 Prompting the 3D Generator

Given the tile image $I(x, y)$ obtained from the 2D image generator in Section 6.3.2, the next goal is to generate a corresponding 3D reconstruction $G(x, y)$ of the tile utilizing an image-to-3D model Φ_{3D}. We opt for using a robust 3D generator and select TRELLIS [Xiang et al. 2024a] due to its good performance, ability to generate both shape and texture, and latent space structure, which will be easy to manipulate for blending as we show later in Section 6.3.4.

Hence, 3D reconstruction amounts to drawing a sample $G(x, y) \sim \Phi_{\text{3D}}(J(x, y))$ from the image-to-3D generator Φ_{3D}. Rather than conditioning on the image $I(x, y)$, we use a pre-processed version $J(x, y)$, as described next.

2D foreground extraction and rebasing. Recall that the image $I(x, y)$ output by the 2D generator of Section 6.3.2 is an image of the tile and its context. Instead, the 3D generator Φ_{3D} expects the input image to focus only on the ob-

ject that needs to be reconstructed, i.e. the new tile. The first step is thus to extract from $I(x, y)$ only the part that corresponds to the new tile, which we do by `rembg` [Gatis 2025] using alpha matting [Brinkmann 2008], as shown in Fig. 6.7.

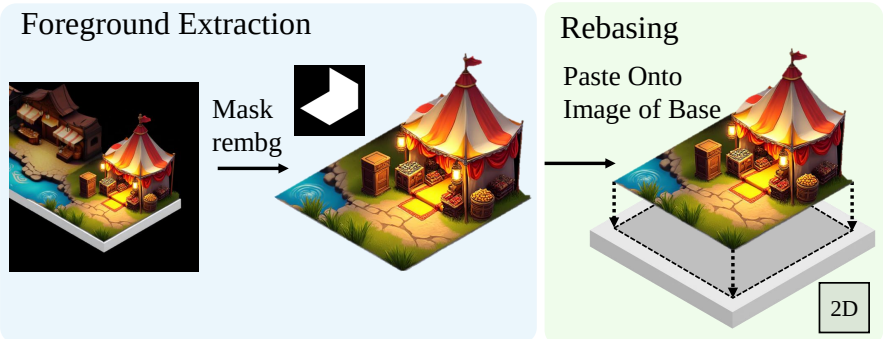


Figure 6.7: *Left:* Isolating the image of the new tile from $I(x, y)$. *Right:* Placing a slightly larger base underneath.

The resulting image is narrowly cropped around the new tile. Similar to Section 6.3.2, we found it beneficial to hallucinate a base for the tile, an operation that we call 'rebasing', as shown in Fig. 6.7. We simply compose the image of the tile with a slightly larger gray slab (in 2D) to obtain $J(x, y)$, which in effect provides a 'frame' for the 3D generator to work with. The base is reconstructed as part of the tile's geometry, which can be used for validation and as a simple-to-detect handle for further 3D processing.

The 'rebased' image $J(x, y)$ is fed to the 3D generator Φ_{3D} to obtain the 3D reconstruction $G(x, y)$ of the tile, which are 3D Gaussian Splats (3DGS). The effect of rebasing on the 3D result is shown in Fig. 6.8.

3D geometric validation. Because the generators are imperfect, we verify the 3D reconstruction $G(x, y)$ to ensure that it is of sufficient quality. If not, we discard it and generate the tile again using a different random seed. To verify the tile, we use a few heuristics that check that the tile's geometry occupies a square region of sufficient size and that the base of the tile has been reconstructed faithfully. Please see Section 6.6.3.

3D post-processing. At this point, we have verified the 3D reconstruction $G(x, y)$ for the tile as a mixture of 3D Gaussian, However, the actual 3D footprint, orientation, and size of the tile are controlled by the 3D generator and are inconsistent. The post-processing utilizes simple heuristics to crop the subset of

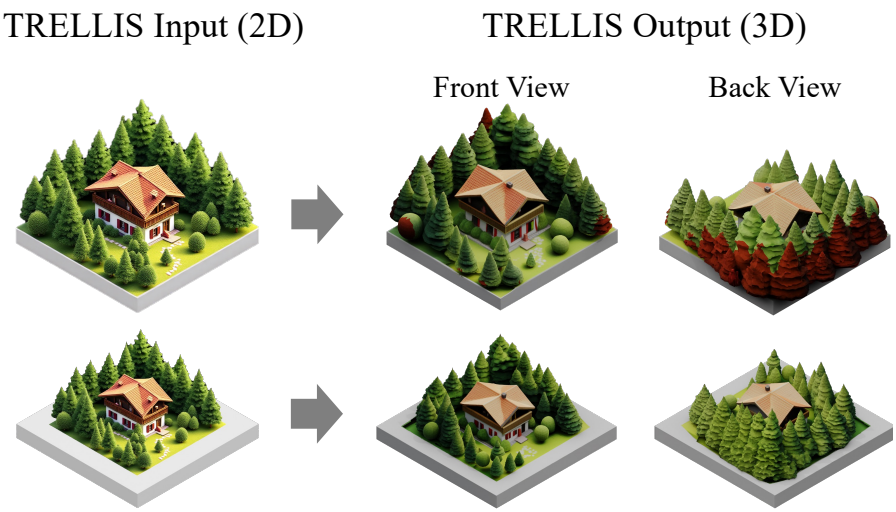


Figure 6.8: *Top:* 3D reconstruction using a tight base. *Bottom:* the same, but with a slightly larger base, which helps to contain the tile's geometry above ground (see the back of the reconstruction), and creates an easy-to-detect 3D base.

3D Gaussians that form the tile proper, removing the extended based, followed by rescaling to fill the full tile, and reorienting the tile so as to face the same way as the 2D image prompt. We explain this in more detail in Section 6.6.4.

6.3.4 3D Blending

At this point of the pipeline, we have reconstructed all 3D tiles $G(x, y)$, $(x, y) \in \mathcal{T}$. Due to the prompting and post-processing steps in Sections 6.3.2 and 6.3.3, the tiles are already approximately aligned and oriented correctly, including being roughly level with the ground. Because the 3D reconstructions are 3D Gaussian Splats, it is easy to simply take their union as the reconstruction of the whole world.

Even so, the boundaries of the tiles may not match perfectly. This is largely due to the fact that TRELLIS does not reconstruct the input images exactly and to the fact that only a single view of each tile is provided to it, which only indirectly controls the reconstruction of the back of the tile. In this section, we thus propose a method to blend the tiles better, ensuring that the 3D world is coherent and continuous.

Although $G(x, y)$ is represented as 3DGS, we can not directly blend them—instead, we can blend the latents, which TRELLIS decodes into 3DGS. Next, we discuss how we blend those latents.

Blending in 2D. To blend the latents of two neighbouring tiles, we first predict the appearance of the boundary between the two tiles. To that end, we place the two 3D tiles next to each other, render a frontal view, and inpaint the middle region of the rendering (Fig. 6.2) with Φ_{2D}. This leads to a well-blended image, which we use to condition for Φ_{3D}.

Blending in 3D. Next, we use Φ_{3D} to blend the latents. We take the latents of the two tiles γ^1 and γ^2 where $\gamma^1, \gamma^2 \in \mathbb{R}^{D \times R \times R \times R}$ are D-dimensional features in the R-sized 3D grid that TRELLIS denoises. We put them together in a new volume γ, where the side where they meet is in the middle:

$$\gamma_{:,x,y,z} = \begin{cases} \gamma^1_{:,x+R/2,y,z}, & \text{if } x < R/2 \\ \gamma^2_{:,x-R/2,y,z}, & \text{if } x \geq R/2. \end{cases}$$

We apply the denoising function Ω, which is the latent denoiser of Φ_{3D}, to the volume γ, but only within the middle region where we have applied the stitch, i.e. for $x \in [R/2-r, R/2+r]$ for some $r < R/2$, while keeping the rest fixed. Formally, we initialise $\tilde{\gamma} \sim \mathcal{N}(0, I)$ and at each denoising step t, we update $\tilde{\gamma}$ as:

$$\tilde{\gamma}_{t+1,:,x,y,z} = \begin{cases} \Omega(\tilde{\gamma}_{t,:,x,y,z}), & \text{if } |x - R/2| \leq r \\ \gamma_{t+1,:,x,y,z}, & \text{otherwise,} \end{cases}$$

where γ_t is obtained by adding noise to the original γ at the corresponding noise level for step t. In practice, we only denoise the second stage of TRELLIS, keeping the occupancy latents of the first stage fixed. The reason for that is that the first stage is at a very low spatial resolution ($R = 16$, compared to $R = 64$ of the second stage), which gives little flexibility for the size of the denoised region r.

Upsampling the latents. Remember that due to the rebasing, $G(x, y)$ contains a 3D base. While we have removed the base in 3DGS space, we have yet to do the same in the latent space. We use same the cuts we applied in 3DGS space to crop the latents, rounding them to account for the discrete nature of the latent voxel grid. In the previous step, however, we assumed that the latents γ^1 and γ^2 have the same spatial resolution, $\gamma \in \mathbb{R}^{D \times R \times R \times R}$. After cropping, this is not the case

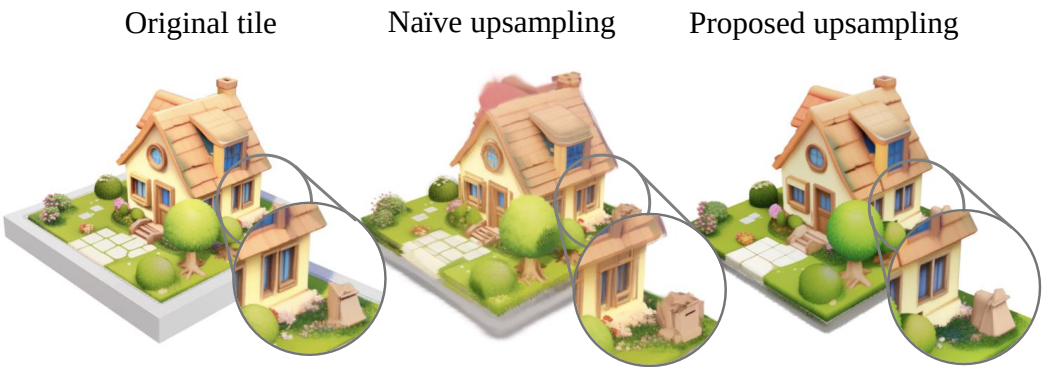


Figure 6.9: **Upsampling sparse latents.** We need to resize or upsample sparse latents in order to stitch them. Due to the sparsity of the latents and the behaviour of the latent decoder, naively resampling in latent space leads to artifacts. Our proposed resizing of the sparse latents better preserves textures and fine structures.

any more if the cuts of neighbouring tiles differ. Thus, similarly to how we resize the tiles' 3DGS to unit size, we have to upsample the now-cropped latents back to the original grid resolution R.

We found that naively upsampling the latents by interpolation leads to poor reconstructions, as shown in Fig. 6.9. We attribute this to the sparse structure of the latents and quirks of the latent decoder of TRELLIS.

We propose the following upsampling scheme. First, we upsample the cropped occupancy volume that TRELLIS predicted to the original resolution $V \in \{0,1\}^{R\times R\times R}$. Next, we denoise *a new set* of latents γ on the upsampled occupancy volume. To preserve the details and textures of the original 3D tile, we render it from multiple views and jointly condition the denoising on all of them. In practice, when denoising with multiple conditioning views, at each timestep, the denoising step is computed as the average denoising step across all views. We show that this upsampling scheme leads to superior reconstructions in Fig. 6.9.

6.4 Experiments

Experimental details We generate the text prompts using ChatGPT o3-mini-high. For the 2D inpainter, we use the Flux ControlNet of [AlimamaCreative 2024].

Human preference. We evaluate human preference for the results generated by our method and those obtained with BlockFusion [Z. Wu et al. 2024]. In particular,

Win Rate (%)				
Overall	Geometry	Exploration	Diversity	Realism
90.9	81.8	90.9	90.9	86.4

Table 6.1: Win rates of our method against BlockFusion. We asked participants to select which scene they prefer overall, as well as which one has better geometry, would be more interesting to explore, is more diverse, and has better realism.



Figure 6.10: *Left:* 2×2 grid generated with our method, where context is taken into account as described in Section 6.3.2 *Right:* Generated with our method using the same prompts, but *not* taking into account context — here, the scale of the buildings is not consistent.

we compare a 'city' scene, showing the entire scene as well as close-up detail views. As seen in Table 6.1, participants ($n = 22$) find our method better overall, with better geometry, realism, and diversity.

6.4.1 Ablations

Here we ablate several components of our approach, showing the importance of each of them.

Building a grid. A naive approach to generating a 3D scene is querying the image generator to produce an image of a large-scale scene (using our 2D image prompt setup) and then obtaining the entire 3D world directly with TRELLIS. To achieve the same level of control provided by our method, the textual prompt needs to be highly detailed and include layout instructions. However, we found neither precise nor abstract prompts to be effective at steering the generations of Flux (for details, see A.4).

2D prompting context. We remove context from neighbouring tiles, as described in Section 6.3.2. Doing that, each tile is sampled independently, and the relative scale between objects is inconsistent, as shown in Fig. 6.10.



Figure 6.11: **Exploring a 3D world.** We show trajectories exploring the 3D worlds we generate. Please see the supplement for videos.

Method	Base Area	Squareness ↑	Completeness ↑
No Rebasing	2271	0.92	0.73
Ours	4096	**1.00**	**1.00**

Table 6.2: Average tile 3D geometry metrics for an approach without rebasing and our method. Rebasing is crucial to ensure a tile is square and its base has been reconstructed faithfully. The metrics we use are the area of the base in voxels, a measure for the 'squareness' of the base, and how many border voxels have been faithfully reconstructed. For details, please refer to the appendix.

Rebasing. To place tiles on a grid, they need to be square (otherwise the grid would be jagged) and their base needs to have been reconstructed faithfully (clearly delimiting where the tile stops). Without rebasing, the geometry generated by TRELLIS might extend beyond the base and makes the tile's 'true' extent difficult to detect, as shown in Fig. 6.8. We ablate the effect of rebasing, using a small 2×2 scene to curb the effect of error accumulation. As seen in Table 6.2, no rebasing causes TRELLIS to generate tiles that are, on average, neither perfectly square nor have a solid border.

Method	LPIPS ↓	SSIM ↑	FID ↓	KID ↓
Naive upsampling	0.5914	0.3093	200.5	0.243
Ours (single frame)	0.3517	0.5149	111.6	0.069
Ours (multi frame)	**0.3212**	**0.5312**	**89.1**	**0.051**

Table 6.3: Perceptual metrics for our methods and the naive approach. Lower values for LPIPS [Richard Zhang et al. 2018], FID [Heusel et al. 2017], and KID [Bińkowski et al. 2018] are better, while higher values for SSIM are better. We see that even using a single conditioning frame leads to better upsampling results, and multiple frames further improve performance.

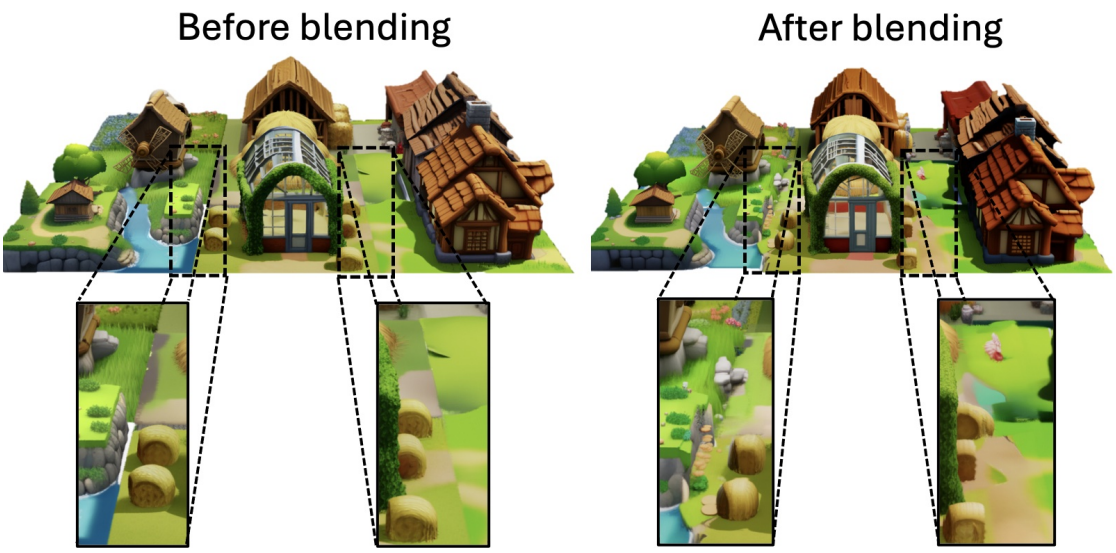


Figure 6.12: *Left:* Tiles before applying the 3D blending step *Right:* After the 3D blending step. We see that where boundaries between tiles were obvious, they are now well-blended.

Latent upsampling. We sample 10 random views each from 200 tiles generated by TRELLIS and compute perceptual metrics in Table 6.3 when upsampling with our proposed upsampling approach in Section 6.3.4 and a naive interpolation. We see that the proposed method leads to better results across a range of metrics, even when using a single conditioning frame.

3D blending. In Fig. 6.12, we generate a scene where we do not apply the 3D blending (Section 6.3.4), resulting in discontinuities between the tiles.

6.4.2 Qualitative Results

We present example scenes generated by our method in Fig. 6.1. Further, we show detail views, highlighting the quality and diversity of the scene. Please see the supplementary material for many more examples.

Exploring a generated world. We can sample trajectories exploring the generated 3D worlds (Fig. 6.11). A skybox has been added for visual effect. Unlike the trajectories generated by world video models [Bruce et al. 2024], ours are guaranteed to be consistent and do not suffer from semantic drift. Different from other systems that only generate a bubble, our method creates spaces sufficiently large to be navigated in a non-trivial manner.

6.5 Conclusion

We have introduced SynCity, an approach to generate diverse, high-quality, and complex 3D worlds with fine-grained control over their layout and appearance. SynCity creates worlds by autoregressively generating tiles on a grid, enabling scalability to arbitrary grid sizes. By accounting for local context and by means of 3D inpainting, the tiles are seamlessly stitched together into coherent scenes. SynCity is flexible: it can either generate worlds from a brief 'world' text prompt or allow control of the individual tiles via tile-specific instructions, all the while maintaining an overall thematic consistency of the generated world. The rich detail of the generated worlds can be fully explored, not restricted to a single '3D bubble' as in many prior works.

We have demonstrated the effectiveness of off-the-shelf generation by utilizing pre-trained language, 2D, and 3D generators through carefully designed prompting strategies and without requiring retraining of any of these components. Nevertheless, we expect that in cases where 3D scene scale data is available, fine-tuning some components would result in further improved results and simplifications in the alignment and rebasing steps of the pipeline. Future work could also consider relaxing the tile structure, for example, by randomly shifting and scaling tiles and using coarse-to-fine modeling to ensure coherent global structure and fine-grained local details.

6.6 Appendix

6.6.1 Language model prompting details

While the prompt p can be constructed manually, an LLM may also be employed. For it to understand the task it is asked to solve, we utilize the following prompt for ChatGPT o3-mini-high:

> Assume you had access to an AI model that can generate small-scale cities on an isometric grid by creating individual tiles. For each of these tiles (identified by their 2D position), a short but expressive text prompt has to be provided. Additionally, a global prompt is used, which provides context, lighting, time of day, as well as the art style.

> The prompts of the tiles can be generic but they might have a semantic connection to neighbouring tiles (such that a river can flow through the city on multiple tiles). The format for the instructions to the AI model is JSON. Consider the following example: `<TEMPLATE>` The art style and perspective mentioned in the prompt should be maintained. The rest may be freely adapted. There are no limits to the setting, the sky is your limit. Now, please generate a 3×3 grid.

In this prompt, a template or a 'seed' prompt is provided. We use a simple JSON file, as exemplified in Fig. 6.13.

```
{
  "tiles": [
    { "prompt": "ancient stone bridge over a stream", "x": 0, "y": 0 },
    { "prompt": "lively stream past mossy banks", "x": 1, "y": 0 },
    { "prompt": "serene pond reflecting moonlight", "x": 0, "y": 1 },
    { "prompt": "bustling medieval market street", "x": 1, "y": 1 }
  ],
  "prompt": "{tile_prompt}, medieval setting, isometric view"
}
```

Figure 6.13: Example JSON file to describe each tile in a 2×2 world.

6.6.2 2D prompting details

In Sec. 3.2, we describe how tiles are generated in the context of those that already exist. There is a special case that we address separately, where context has to be bootstrapped, namely tiles $\mathcal{L} := \{(x, y) \in \mathcal{T} : x = 0 \wedge y > 0\}$. Due to our build order and the trimming of obstructing 3D geometry, these tiles might lack sufficient contextual cues. As a remedy, we temporarily provide context with a previously generated tile: For a tile $(0, y) \in \mathcal{L}$, we duplicate the tile $(0, y - 1) \in \mathcal{T}$ and place the copy at position $(-1, y)$. During inpainting, this tile serves to provide context in terms of scale and general appearance. Once inpainting is completed, this copy is removed.

6.6.3 3D geometry validation details

To perform geometric validation, we note first that TRELLIS is a two-stage method, and in the first stage (before the 3D Gaussian mixture is output) it produces an occupancy volume $V \in \{0, 1\}^{R \times R \times R}$ where $R = 64$ is the resolution

of the grid. This occupancy volume captures the rough 3D geometry of the tile. We check that this 3D volume conforms to the desired geometry.

First, we test whether the reconstructed tile is supported by a square by computing its 2D rectangular footprint and ensuring that the latter is sufficiently large and isotropic. Then, we check that the base that we have added in 2D in the 'rebasing' step has been faithfully reconstructed. To this end, let the *base template* V_B be a voxel grid containing the base of the tile, built adaptively as detailed below. We discard the tile if $(V \cdot V_B)/(V_B \cdot V_B) < \beta = 0.95$, where $\cdot$ denotes the inner product of tensors.

Recall that we validate the voxel grid V by testing the area and aspect ratio of its horizontal footprint. To this end, let (u, v, w) index the $R \times R \times R$ voxel grid, where u and v corresponds to world directions x and y. Let $(u_{\min}, u_{\max}, v_{\min}, v_{\max},)$ be the bounding box containing all the active voxels at height w.[1] Let $\text{ext}_u = \max\{0, 1 + u_{\max} - u_{\min}\}$ the width of the bounding box and ext_v its height. We discard the tile if the area is too small, i.e. if $\text{ext}_u \cdot \text{ext}_v < (R/2)^2$. We also discard it if it is not square enough, i.e. if $\min\{\text{ext}_u, \text{ext}_v\}/\max\{\text{ext}_u, \text{ext}_v\} < \alpha = 0.95$.

To define the 3D tile base template V_B, let $u_{\min}(w)$ be the minimum u of the bounding box that contains the volume slice at height w, and define $u_{\max}(w)$ and so on in a similar manner, so for instance $u_{\min}(w) = \min\{u : \exists v : V(u, v, w) = 1\}$. Let w^* be the height at which the base is the largest, i.e. $w^* = \text{argmax}_w \, \text{ext}_u(w) \cdot \text{ext}_v(w)$. Then, V_B is the indicator function of the voxels (u, v, w) such that $w = w^*$ and

$$\max\left\{\frac{|2u - u_{\max}(w^*) - u_{\min}(w^*)|}{u_{\max}(w^*) - u_{\min}(w^*)}, \; \frac{|2v - v_{\max}(w^*) - v_{\min}(w^*)|}{v_{\max}(w^*) - v_{\min}(w^*)}\right\} = 1.$$

Note that the template V_B is constructed adaptively to match the input tile V.

6.6.4 3DGS post-processing details

3D cropping, resizing and centering. Given the 3D Gaussian mixture $G(x, y)$ initially output by the 3D generator, we first identify the extent of the tile 'proper' (discounting the extended base). We consider the xy footprint of the tile (i.e. we look at the tile from above) and seek to identify four cuts (from the left, right, top, and bottom) that define an axis-aligned rectangle strictly containing the tile. For example, to determine the location of the left cut x^*, we consider slices $V_x = \{(x', y', z') \in \mathbb{R}^3 : x - \delta \leq x' < x + \delta\}$. We find the 3D Gaussians whose

[1] So for instance $u_{\min} = \min\{u : \exists v, w : V(u, v, w) = 1\}$.

centers falls within V_x and compute their average color c_x. Then, we compute the distance $d(x) = \|c_x - c_{x_{\min}}\|$ where $c_{x_{\min}}$ is the average color of the leftmost slice (used as a reference). We set $x^* = \min\{x : d(x) > \tau\}$ where τ is a threshold, which corresponds to the slice that transition from the 'background' color to something else.

We find in this manner the four cuts, keep only the Gaussians contained in the resulting rectangular footprint, and recenter and resize this footprint to fill the standard tile size.

Additionally, the base allows us to figure out the position of the tile's surface: As TRELLIS centers objects vertically, the ground surface level of any two tiles will vary. We use the average height of the tile's four corners to determine the position of the surface, allowing us to align it with others.

3D reorientation. We also note that TRELLIS generates the 3D object with an arbitrary orientation with respect to the input image $\tilde{I}(x, y)$. However, the tile must be inserted with the correct orientation in the 3D world a otherwise the continuity between tiles, which the inpainting method of Fig. 6.4 encourages, will be lost. In practice, the ambiguity is limited to 90-degree rotations around the vertical axis and is very easy to remove.[2] To do so, we test four possible 90-degree rotations of the tile around the vertical axis, rendering the corresponding views and comparing them to $\tilde{I}(x, y)$ using the LPIPS [Richard Zhang et al. 2018] loss. The minimizer is taken as the correct orientation.

6.6.5 Ablation details

Building a Grid. In the following, we present results from our experiments attempting to generate a large scene non-iteratively. Here, we generate single image with Flux that is used as conditioning for TRELLIS to directly create the desired scene.

In the first set of experiments, we do not use our 2D prompting design. To obtain an isolated 3D object that can be generated by TRELLIS, we use prompts with the prefix "a 3d object of". We show those results in Fig. 6.14. While the generated

[2]This is likely due to the implicit bias in the TRELLIS training set that consists of synthetic 3D objects which are almost invariably axis aligned.

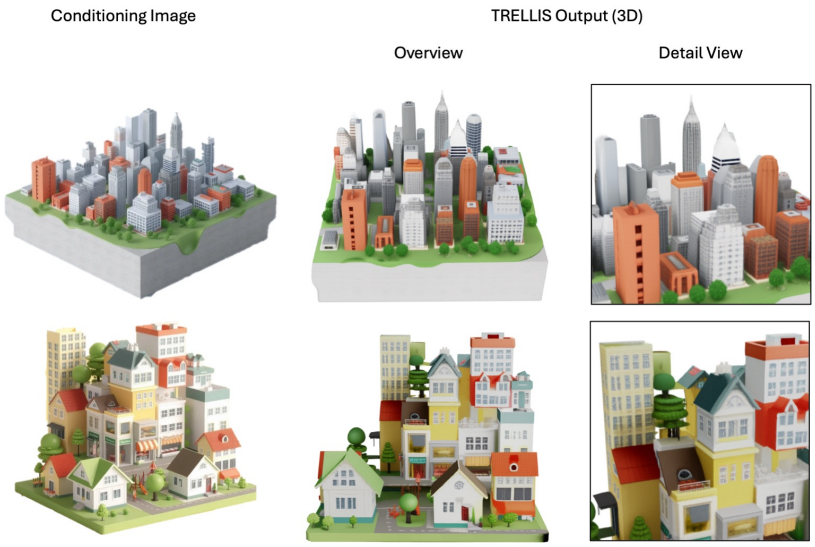


Figure 6.14: **Non-iterative city building.** We obtain conditioning images generated by Flux (left) and directly use them to build a large-scale scene with TRELLIS (center). While the generated 3D structures are visually appealing, the level of detail (right) is very limited. The first row used generic prompting for the conditioning image ("a city scene on top of a base"), whereas the second row uses a more involved prompt with an explicit layout (e.g., "a house in the bottom left corner", "a pharmacy in the top right corner").

objects are visually appealing, they have several limitations: (i) The resolution of the conditioning image and the 3D structures TRELLIS can generate is limited. Therefore, this approach is not scalable to arbitrarily large scenes. (ii) Due to the lack of perfect control over the base structure, the result cannot be easily extended or edited. (iii) The layout instructions are mostly ignored, thus severely limiting the level of control over the generation.

For the second set of experiments, we use our 2D prompting design along with the Flux ControlNet for inpainting (Fig. 6.15). However, with this setup, the quality of the results is not improved. The layout instructions in the prompt are mostly ignored, again.

Querying Flux to generate large-scale scenes directly has not been successful in our experiments, prompting the need for our grid-based method that allows fine-grained layout and appearance control for each tile.

6.6.6 Additional qualitative results.

In Figs. 6.16 to 6.19, we show additional results of our method. As we leverage a pre-trained 2D image generator trained on a very large dataset, we are able to

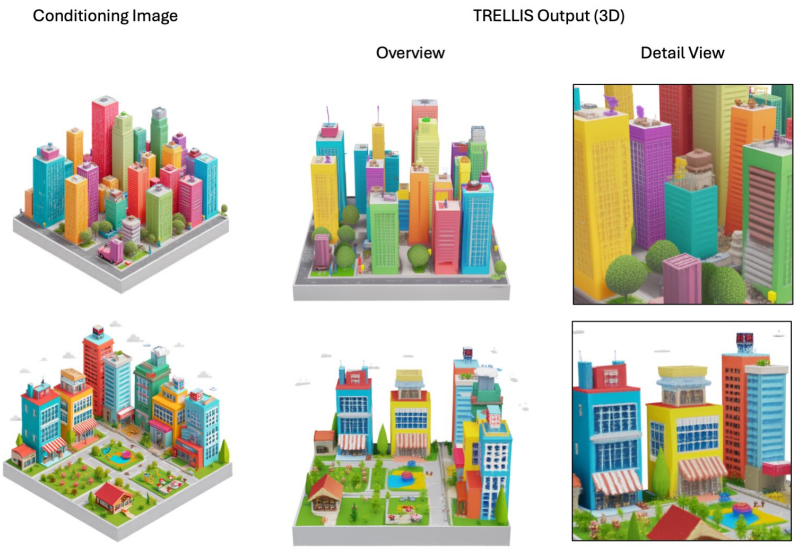


Figure 6.15: **Non-iterative city building (with our 2D prompting).** We obtain conditioning images generated by Flux (left) and directly use them to build a large-scale scene with TRELLIS (center). Despite the initial visual appeal, the structures lack in detail. The first row used generic prompting for the conditioning image ("a vibrant city scene"), whereas the second row uses a more involved prompt with an explicit layout (e.g., "a house in the bottom left corner", "a pharmacy in the top right corner").

generate highly diverse scenes. Thanks to our fine-grained control at the tile level, we can generate interesting patterns, such as a transition in seasons across a grid (observe the largest grid in Fig. 6.19).

6.6.7 Limitations

While our method allows creating large and diverse scenes, there are some limitations to be addressed in future work.

Atomic tiles. Although we inpaint tiles conditioned on their surroundings, they remain individual units. While structures can be created that span across multiple tiles, this requires harmonious cooperation between Flux and TRELLIS.

Use of heuristics. To determine the ground surface height for each tile and removing the base we added during rebasing, we employ heuristics. We have designed these carefully with fallback mechanisms but they are not infallible.

Inherited limitations. As our method builds on top of Flux and TRELLIS, their limitations also apply to ours. During our experiments, we have observed

that despite good inpainting results, TRELLIS at times only vaguely adheres to the conditioning image in terms of appearance and in particular color. Thus, transitions between tiles might not look perfectly smooth (even though the conditioning image had the right foundation).

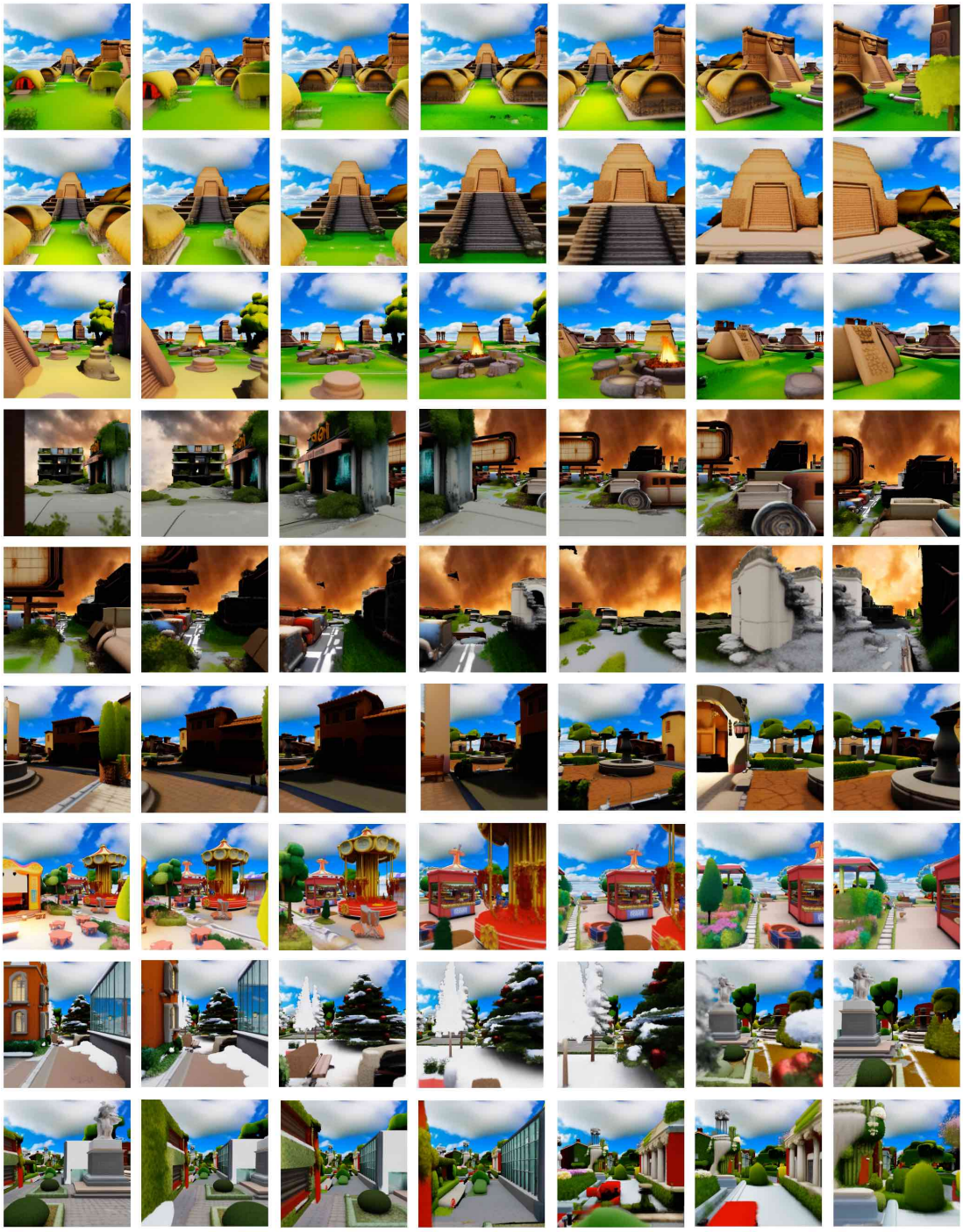


Figure 6.16: **Exploring a 3D world.** We show trajectories exploring the 3D worlds we generate. A sky box has been added for visual effect.



Figure 6.17: **Generated scenes.** We show scenes that we generate with the same prompts, but using different seed for the 2D inpainting.



Figure 6.18: **Generated scenes.**



Figure 6.19: **Generated scenes.** We show larger scenes, where *Top:* the scene goes from winter to spring to summer to autumn, and *Bottom:* where the scenery changes from city to village.

Chapter 7

Discussion

7.1 Summary and Impact

In this section, we summarize the contributions of this thesis and discuss their broader impact. Finally, we review some ideas for future work that builds upon this thesis.

7.1.1 Unsupervised Correspondences

In Chapter 2, we presented the paper "**Learning Universal Semantic Correspondences with No Supervision and Automatic Data Curation**". We identified a shortcoming of DINO features, which despite being state-of-the-art in unsupervised semantic correspondences, lack spatial precision. We proposed a method that distills them, while incorporating a geometric prior in order to make them more localized. Additionally, we propose a framework where the query features are adapted by the appearance of the target image, "predicting" what the corresponding target feature would be. Next, we proposed to train using a large uncurated dataset, removing the requirement of prior works to use human-annotated image pairs. We found that the features learnt in that way are more general and can be applied to different evaluation datasets, without retraining on them like prior works.

The work in Chapter 2 has influenced a number of follow up works. [Junyi Zhang et al. 2023b] distil geometry-aware features from foundation models, specifically for telling left from right and [Mariotti et al. 2024] supplement foundation model

features with 3D geometric priors. The work has also been referenced in methods for point tracking in videos [I. H. Kim et al. 2025; Tumanyan et al. 2024], where DINO features are adapted for that purpose, and change detection [F. Zhou et al. 2025], which was inspired by our appearance adaptation scheme.

In Chapter 3, we presented the paper "**SHIC: Shape-Image Correspondences with No Keypoint Supervision**". We began by reducing the task of image-to-3D correspondences to that of image-to-image correspondences by using 2D renders of the 3D shapes. Next, we found that off-the-shelf semantic correspondence methods can already find high-quality correspondences between a natural image and a non-photorealistic render of a template mesh. We also proposed to aggregate such correspondences from multiple views, solving issues of left/right ambiguity and occlusions. Finally, we used the discovered correspondences with that zero-shot approach to train a feedforward method, which is faster and more accurate. We found that the models we learn in that manner outperform the supervised state-of-art, which requires hundreds of thousands of manual annotations.

The paper in Chapter 3 has inspired several follow-up works. [W. Guo et al. 2025], uses pseudo-correspondences from image-to-image correspondences, lifted to 3D, to train a 6D pose estimation model. [Uzolas et al. 2025] render views of 3D objects and aggregate 2D foundation model features on the object's surface, in order to do 3D shape matching.

7.1.2 Visual prompting

In Chapter 4, we presented the paper "**What does CLIP know about a red circle? Visual prompt engineering for VLMs**". In that work, we discovered an emergent ability of CLIP, where by overlaying a red circle on an image, we draw (pun intended) the vision encoder's attention to that region. We termed the process of overlaying manually designed marker over images "visual prompt engineering" and showcase the power of that method by achieving state-of-the-art in unsupervised referring expressions comprehension. We thoroughly explored different marker shapes and colors, and model sizes and pertaining datasets. We found that red circles perform best by far, and this ability only emerges in the largest models, trained on the largest datasets. Finally, we discovered an alarming bias of CLIP and a drawback of using such models, where people that are annotated

with red circles tend to be classified as murderers or victims.

The work in Chapter 4 has inspired a large number of follow-up papers. Similar emergent behaviours were observed in multimodal LLMs, such as GPT-4 [Zhengyuan Yang et al. 2023] and several open-source multimodal VLMs have been fine-tuned to reason about visual markers [Jiannan Wu et al. 2024; M. Cai et al. 2024; Lei Li et al. 2023]. Other visual prompt engineering strategies were developed, such as using contours [Lingfeng Yang et al. 2023] or learnable prompts [Rezaei et al. 2024]. [Jinheng Xie et al. 2024] propose an object detection method, where the location of the visual prompt is iteratively optimised for a given text prompt. Overlaying images with visual markers such as red circles is now a common way of prompting VLMs, and has been used in a number of applications, such as medical image analysis [Denner et al. 2024], audio descriptions generation [Junyu Xie et al. 2024], 3D generation [Minghao Chen et al. 2024], trajectory selection [Daeun Song et al. 2024], robotics [Nasiriany et al. 2024], traffic analysis [Z. Duan et al. 2024], video reasoning [B. Liu et al. 2024], 3D reasoning [Dingning Liu et al. 2023], gaussian splatting [Wanhua Li et al. 2025].

In Chapter 5 we presented the paper "**Highlight: Learning Visual Prompts for Vision-Language Models**". In that work, we explored automatically learning a visual prompt, instead of using a manually engineered one. We developed several strategies to optimise a visual prompt, with and without using supervised data. The visual prompts we learn outperform the manually engineered ones. Interestingly, we find that the automatically learnt visual prompts look like red contours, strikingly resembling the manually engineered red circle.

7.1.3 3D Scene Generation

In Chapter 6 we presented the paper "**SynCity: Training-Free Generation of 3D Worlds**". In this work, we use similar ideas as in Chapter 4 to elicit useful behaviours learn from data through prompt engineering, but for generative models. We apply this to both 2D and 3D generators in order to generate 3D scenes from a text prompt, a task not attempted yet from prior works. Since its release, this project has received a total of more than half a million impressions on social media and 600 stars on GitHub. The majority of this interest comes from 3D artists and game designers, showing the practical utility of the work.

7.2 Future Work

Lastly, we outline promising future research directions.

More modalities Recent progress in fields such as video generation [Kong et al. 2024; Zhuoyi Yang et al. 2024] presents a great opportunity to develop unsupervised methods for reasoning or generation beyond a single image. A possible area of interest could be predicting temporal dynamics in a video, e.g. tracking points or objects in a video using features from a video generation model. Likewise, a video generation model could be used as a motion prior in order to animate arbitrary articulated 3D shapes, or reason about motion.

Visual prompting Using visual prompts could present an opportunity to train multimodal LLMs to handle various vision tasks they currently struggle at. For example, reasoning about long videos is notoriously difficult and expensive. By overlaying motion cues, such as tracks from a point tracker like CoTracker [Karaev et al. 2024], one could explicitly encode motion and enable long-range reasoning about motion. Similarly, one could train a multimodal LLM using 3D-aware prompts such as pointmaps [Shuzhe Wang et al. 2024] in addition to images, to enable 3D reasoning or planning.

7.3 Conclusion

In this thesis, we explored unsupervised scene understanding with foundation models through the lens of emergent properties and behaviours of these models. We showed how we can repurpose features from large pretrained models to solve low-level vision tasks, such as point correspondences, in both 2D and 3D. We also showed how we can elicit useful behaviour from foundation models by treating them as black boxes and tweaking their inputs, which enables novel use cases in both understanding and generation. Beyond these immediate applications, this opens up opportunities for new ways of repurposing these models.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. (2023). "Gpt-4 technical report". In: *arXiv preprint arXiv:2303.08774*.

Sandhini Agarwal, Gretchen Krueger, Jack Clark, Alec Radford, Jong Wook Kim, and Miles Brundage (2021). "Evaluating clip: towards characterization of broader capabilities and downstream implications". In: *arXiv preprint arXiv:2108.02818*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. (2022). "Flamingo: a visual language model for few-shot learning". In: *Advances in neural information processing systems*.

AlimamaCreative (2024). *FLUX-ControlNet-Inpainting*. https://github.com/alimama-creative/FLUX-Controlnet-Inpainting. GitHub repository.

Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel (2021a). "Deep ViT Features as Dense Visual Descriptors". In: *CoRR*.

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

Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Michael Rabbat, and Nicolas Ballas (2022). "Masked Siamese Networks for Label-Efficient Learning". In: *arXiv preprint arXiv:2204.07141*.

Mehmet Aygun and Oisin Mac Aodha (2022). "Demystifying Unsupervised Semantic Correspondence Estimation". In: *European Conference on Computer Vision (ECCV)*.

Hessam Bagherinezhad, Maxwell Horton, Mohammad Rastegari, and Ali Farhadi (2018). "Label refinery: Improving imagenet classification through label progression". In: *arXiv preprint arXiv:1805.02641*.

Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola (2022). "Visual prompting: Modifying pixel space to adapt pre-trained models". In: *arXiv preprint arXiv:2203.17274*.

Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei A Efros (2022). "Visual prompting via image inpainting". In: *arXiv preprint arXiv:2209.00647*.

Miguel Angel Bautista, Pengsheng Guo, Samira Abnar, Walter Talbott, Alexander Toshev, Zhuoyuan Chen, Laurent Dinh, Shuangfei Zhai, Hanlin Goh, Daniel Ulbricht, et al. (2022). "Gaudi: A neural architect for immersive 3d scene generation". In: *Advances in Neural Information Processing Systems*.

Hugo Berg, Siobhan Mackenzie Hall, Yash Bhalgat, Wonsuk Yang, Hannah Rose Kirk, Aleksandar Shtedritski, and Max Bain (2022). "A prompt array keeps the bias away: Debiasing vision-language models with adversarial learning". In: *arXiv preprint arXiv:2203.11933*.

Benjamin Biggs, Oliver Boyne, James Charles, Andrew Fitzgibbon, and Roberto Cipolla (2020). "Who left the dogs out?: 3D animal reconstruction with expectation maximization in the loop". In: *ECCV*.

Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton (2018). "Demystifying mmd gans". In: *arXiv preprint arXiv:1801.01401*.

Bloomberg News (Dec. 2023). "Large AI Dataset Has Over 1,000 Child Abuse Images, Researchers Find". In: *Bloomberg*. Accessed: 2025-04-23. URL: https://www.bloomberg.com/news/articles/2023-12-20/large-ai-dataset-has-over-1-000-child-abuse-images-researchers-find.

Ron Brinkmann (2008). *The art and science of digital compositing: Techniques for visual effects, animation and motion graphics*. Morgan Kaufmann.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. (2023). "Rt-2: Vision-language-action models transfer web knowledge to robotic control". In: *arXiv preprint arXiv:2307.15818*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. (2020). "Language models are few-shot learners". In: *Advances in neural information processing systems*.

Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer (2017). "Adversarial patch". In: *arXiv preprint arXiv:1712.09665*.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. (2024). "Genie: Generative interactive environments". In: *Forty-first International Conference on Machine Learning*.

Sébastien Bubeck, Varun Chadrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. (2023). *Sparks of artificial general intelligence: Early experiments with gpt-4*.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil (2006). "Model compression". In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Mu Cai, Haotian Liu, Siva Karthik Mustikovela, Gregory P Meyer, Yuning Chai, Dennis Park, and Yong Jae Lee (2024). "Vip-llava: Making large multimodal models understand arbitrary visual prompts". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Shengqu Cai, Eric Ryan Chan, Songyou Peng, Mohamad Shahbazi, Anton Obukhov, Luc Van Gool, and Gordon Wetzstein (2023). "DiffDreamer: Towards Consistent Unsupervised Single-view Scene Extrapolation with Conditional Diffusion Models". In: *Proceedings of the International Conference on Computer Vision (ICCV)*.

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh (2017). "Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields". In: *CVPR*.

Andrea Caraffa, Davide Boscaini, Amir Hamza, and Fabio Poiesi (2024). "Freeze: Training-free zero-shot 6d pose estimation with geometric and vision foundation models". In: *European Conference on Computer Vision*. Springer.

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

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

Lucy Chai, Richard Tucker, Zhengqi Li, Phillip Isola, and Noah Snavely (2023). "Persistent nature: A generative model of unbounded 3d worlds". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

ChatGPT (n.d.). https://chat.openai.com/.

Hila Chefer, Shir Gur, and Lior Wolf (2021). "Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Jianchun Chen, Lingjing Wang, Xiang Li, and Yi Fang (2019). "Arbicon-Net: Arbitrary Continuous Geometric Transformation Networks for Image Registration". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2019/file/56f9f88906aebf4ad985aaec7fa01313-Paper.pdf.

Kan Chen, Rama Kovvuri, and Ram Nevatia (2017). "Query-guided regression network with context policy for phrase grounding". In: *Proceedings of the IEEE International Conference on Computer Vision*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. (2021). "Evaluating large language models trained on code". In: *arXiv preprint arXiv:2107.03374*.

Minghao Chen, Roman Shapovalov, Iro Laina, Tom Monnier, Jianyuan Wang, David Novotny, and Andrea Vedaldi (2024). "PartGen: Part-level 3D Generation and Reconstruction with Multi-View Diffusion Models". In: *arXiv preprint arXiv:2412.18608*.

Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia (2023). "Fantasia3D: Disentangling Geometry and Appearance for High-quality Text-to-3D Content Creation". In: *arXiv.cs*.

Yi-Wen Chen, Yi-Hsuan Tsai, Tiantian Wang, Yen-Yu Lin, and Ming-Hsuan Yang (2019). "Referring expression object segmentation with caption-aware consistency". In: *arXiv preprint arXiv:1910.04748*.

Zhaoxi Chen, Guangcong Wang, and Ziwei Liu (2023). "Scenedreamer: Unbounded 3d scene generation from 2d image collections". In: *IEEE transactions on pattern analysis and machine intelligence*.

Zezhou Cheng, Jong-Chyi Su, and Subhransu Maji (2021). "On equivariant and invariant learning of object landmark representations". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev (2022). "Reproducible scaling laws for contrastive language-image learning". In: *arXiv preprint arXiv:2212.07143*.

Seokju Cho, Sunghwan Hong, Sangryul Jeon, Yunsung Lee, Kwanghoon Sohn, and Seungryong Kim (2021). "Cats: Cost aggregation transformers for visual correspondence". In: *Advances in Neural Information Processing Systems*.

Seokju Cho, Sunghwan Hong, and Seungryong Kim (2022). "CATs++: Boosting Cost Aggregation with Convolutions and Transformers". In: *arXiv preprint arXiv:2202.06817*.

Sumit Chopra, Raia Hadsell, and Yann LeCun (2005). "Learning a Similarity Metric Discriminatively, with Application to Face Verification". In: IEEE.

Ching-Yao Chuang, Varun Jampani, Yuanzhen Li, Antonio Torralba, and Stefanie Jegelka (2023). "Debiasing vision-language models via biased prompts". In: *arXiv preprint arXiv:2302.00070*.

Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee (2023). "Luciddreamer: Domain-free generation of 3d gaussian splatting scenes". In: *arXiv preprint arXiv:2311.13384*.

Volkan Cirik, Taylor Berg-Kirkpatrick, and Louis-Philippe Morency (2018). "Using syntax to ground referring expressions in natural images". In: *Proceedings of the AAAI conference on artificial intelligence*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. (2021). "Training verifiers to solve math word problems". In: *arXiv preprint arXiv:2110.14168*.

Dana Cohen-Bar, Elad Richardson, Gal Metzer, Raja Giryes, and Daniel Cohen-Or (2023). "Set-the-Scene: Global-Local Training for Generating Controllable NeRF Scenes". In.

Deemos (2024). *Rodin Text-to-3D Gen-1 (0525) V0.5*. URL: https://hyperhuman.deemos.com/rodin.

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. (2023a). "Objaverse-xl: A universe of 10m+ 3d objects". In: *arXiv preprint arXiv:2307.05663*.

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi (2023b). "Objaverse-XL: A Universe of 10M+ 3D Objects". In: *CoRR*.

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.

Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li (2021). "Transvg: End-to-end visual grounding with transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Stefan Denner, Markus Bujotzek, Dimitrios Bounias, David Zimmerer, Raphael Stock, Paul F Jäger, and Klaus Maier-Hein (2024). "Visual Prompt Engineering for Medical Vision Language Models in Radiology". In: *arXiv preprint arXiv:2408.15802*.

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*.

Zi-Yi Dou, Aishwarya Kamath, Zhe Gan, Pengchuan Zhang, Jianfeng Wang, Linjie Li, Zicheng Liu, Ce Liu, Yann LeCun, Nanyun Peng, et al. (2022). "Coarse-to-fine vision-language pre-training with fusion in the backbone". In: *arXiv preprint arXiv:2206.07643*.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. (2023). "Palm-e: An embodied multimodal language model". In: *arXiv preprint arXiv:2303.03378*.

Zhizhao Duan, Hao Cheng, Duo Xu, Xi Wu, Xiangxie Zhang, Xi Ye, and Zhen Xie (2024). "Cityllava: Efficient fine-tuning for vlms in city scenario". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Niladri Shekhar Dutt, Sanjeev Muralikrishnan, and Niloy J. Mitra (June 2024). "Diffusion 3D Features (Diff3F): Decorating Untextured Shapes with Distilled Semantic Features". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Paul Engstler, Luke Melas-Kyriazi, Christian Rupprecht, and Iro Laina (2023). "Understanding self-supervised features for learning unsupervised instance segmentation". In: *arXiv preprint arXiv:2311.14665*.

Paul Engstler, Aleksandar Shtedritski, Iro Laina, Christian Rupprecht, and Andrea Vedaldi (2025a). "SynCity: Training-Free Generation of 3D Worlds". In: *Arxiv*.

Paul Engstler, Andrea Vedaldi, Iro Laina, and Christian Rupprecht (2025b). “Invisible Stitch: Generating Smooth 3D Scenes with Depth Inpainting”. In: *Proceedings of the International Conference on 3D Vision (3DV)*.

Sepideh Esmaeilpour, Bing Liu, Eric Robertson, and Lei Shu (2021). “Zero-shot open set detection by extending CLIP”. In: *arXiv preprint arXiv:2109.02748*.

Pedro F. Felzenszwalb, David A. McAllester, and Deva Ramanan (2008). “A discriminatively trained, multiscale, deformable part model”. In: *CVPR*.

Rafail Fridman, Amit Abecasis, Yoni Kasten, and Tali Dekel (2023). “SceneScape: Text-Driven Consistent Scene Generation”. In: *CoRR*.

Jensen Gao, Bidipta Sarkar, Fei Xia, Ted Xiao, Jiajun Wu, Brian Ichter, Anirudha Majumdar, and Dorsa Sadigh (2023). “Physically grounded vision-language models for robotic manipulation”. In: *arXiv preprint arXiv:2309.02561*.

Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T. Barron, and Ben Poole (2024). “CAT3D: Create Anything in 3D with Multi-View Diffusion Models”. In: *arXiv preprint*.

Daniel Gatis (Feb. 2025). *rembg*. Version 2.0.63. URL: https://github.com/danielgatis/rembg.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge (June 2016). “Image Style Transfer Using Convolutional Neural Networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos (2018). “DensePose: Dense Human Pose Estimation in the Wild”. In: *CVPR*.

Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos (2018). “Densepose: Dense human pose estimation in the wild”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Weijie Guo, Guofeng Zhang, Wufei Ma, and Alan Yuille (2025). “DINeMo: Learning Neural Mesh Models with no 3D Annotations”. In: *arXiv preprint arXiv:2503.20220*.

Zixian Guo, Bowen Dong, Zhilong Ji, Jinfeng Bai, Yiwen Guo, and Wangmeng Zuo (2022). “Texts as Images in Prompt Tuning for Multi-Label Image Recognition”. In: *arXiv preprint arXiv:2211.12739*.

Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce (2016). “Proposal Flow”. In: *CVPR*.

Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce (2017). “Proposal flow: Semantic correspondences from object proposals”. In: *IEEE transactions on pattern analysis and machine intelligence*.

Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman (2022). “Unsupervised Semantic Segmentation by Distilling Feature Correspondences”. In: *arXiv preprint arXiv:2203.08414*.

Hasan Abed Al Kader Hammoud, Hani Itani, Fabio Pizzati, Philip Torr, Adel Bibi, and Bernard Ghanem (2024). “SynthCLIP: Are We Ready for a Fully Synthetic CLIP Training?” In: *arXiv preprint arXiv:2402.01832*.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick (2021). “Masked Autoencoders Are Scalable Vision Learners”. In.

Eric Hedlin, Gopal Sharma, Shweta Mahajan, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi (2023). “Unsupervised Semantic Correspondence Using Stable Diffusion”. In: *arXiv.cs*. eprint: 2305.15581.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter (2017). “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”. In: *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*.

Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner (2023). “Text2Room: Extracting Textured 3D Meshes from 2D Text-to-Image Models”. In: *Proceedings of the International Conference on Computer Vision (ICCV)*.

Sunghwan Hong, Seokju Cho, Seungryong Kim, and Stephen Lin (2022). “Integrative Feature and Cost Aggregation with Transformers for Dense Correspondence”. In: *arXiv preprint arXiv:2209.08742*.

Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell (2016). “Natural language object retrieval”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao (2024). “2d gaussian splatting for geometrically accurate radiance fields”. In: *ACM SIGGRAPH 2024 conference papers*.

Shuaiyi Huang, Qiuyue Wang, Songyang Zhang, Shipeng Yan, and Xuming He (2019). “Dynamic Context Correspondence Network for Semantic Alignment”. In: *Proceedings of the IEEE International Conference on Computer Vision*.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch (2022). “Language models as zero-shot planners: Extracting actionable knowledge for embodied agents”. In: *International Conference on Machine Learning*. PMLR.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. (2021). "Perceiver io: A general architecture for structured inputs & outputs". In: *arXiv preprint arXiv:2107.14795*.

Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi (2018). "Unsupervised Learning of Object Landmarks through Conditional Image Generation". In.

Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi (2020). "Self-supervised Learning of Interpretable Keypoints from Unlabelled Videos". In.

Sangryul Jeon, Seungryong Kim, Dongbo Min, and Kwanghoon Sohn (2018). "Parn: Pyramidal affine regression networks for dense semantic correspondence". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Sangryul Jeon, Dongbo Min, Seungryong Kim, Jihwan Choe, and Kwanghoon Sohn (2020). "Guided semantic flow". In: *European Conference on Computer Vision*. Springer.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim (2022). "Visual prompt tuning". In: *arXiv preprint arXiv:2203.12119*.

Haojun Jiang, Yuanze Lin, Dongchen Han, Shiji Song, and Gao Huang (2022). "Pseudo-q: Generating pseudo language queries for visual grounding". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi (2021). "Cotr: Correspondence transformer for matching across images". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei (2016). "Perceptual losses for real-time style transfer and super-resolution". In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer.

Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie (2022). "Prompting visual-language models for efficient video understanding". In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*. Springer.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion (2021). "Mdetr-modulated detection for end-to-end multi-modal understanding". In: *Proceedings of the IEEE/CVF international conference on computer vision*.

Angjoo Kanazawa, David W. Jacobs, and Manmohan Chandraker (2016). "WarpNet: Weakly Supervised Matching for Single-View Reconstruction". In: *CVPR*.

Nikita Karaev, Ignacio Rocco, Ben Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht (2024). "CoTracker: It is Better to Track Together". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Kimmo Karkkainen and Jungseock Joo (2021). "FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age for Bias Measurement and Mitigation". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.

Tejan Karmali, Abhinav Atrishi, Sai Sree Harsha, Susmit Agrawal, Varun Jampani, and R Venkatesh Babu (2022). "LEAD: Self-Supervised Landmark Estimation by Aligning Distributions of Feature Similarity". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg (2014). "Referitgame: Referring to objects in photographs of natural scenes". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis (2023). "3D Gaussian Splatting for Real-Time Radiance Field Rendering". In.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li (2011). "Novel dataset for fine-grained image categorization: Stanford dogs". In: *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*. Citeseer.

Inès Hyeonsu Kim, Seokju Cho, Jiahui Huang, Jung Yi, Joon-Young Lee, and Seungryong Kim (2025). "Exploring Temporally-Aware Features for Point Tracking". In: *arXiv preprint arXiv:2501.12218*.

Jiwon Kim, Kwangrok Ryoo, Junyoung Seo, Gyuseong Lee, Daehwan Kim, Hansang Cho, and Seungryong Kim (2022). "Semi-Supervised Learning of Semantic Correspondence with Pseudo-Labels". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Seungryong Kim, Stephen Lin, Sang Ryul Jeon, Dongbo Min, and Kwanghoon Sohn (2018). "Recurrent transformer networks for semantic correspondence". In: *Advances in neural information processing systems*.

Seungryong Kim, Dongbo Min, Somi Jeong, Sunok Kim, Sangryul Jeon, and Kwanghoon Sohn (2019). "Semantic attribute matching networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Wonjae Kim, Bokyung Son, and Ildoo Kim (2021). "Vilt: Vision-and-language transformer without convolution or region supervision". In: *International conference on machine learning*. PMLR.

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

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick (2023b). "Segment Anything". In: *arXiv:2304.02643*.

Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann (2022). "Decomposing NeRF for Editing via Feature Field Distillation". In: *arXiv preprint arXiv:2205.15585*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa (2022). "Large language models are zero-shot reasoners". In: *Advances in neural information processing systems*.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. (2024). "Hunyuanvideo: A systematic framework for large video generative models". In: *arXiv preprint arXiv:2412.03603*.

Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi (2019). "Pifpaf: Composite fields for human pose estimation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Alexander Krull, Tim-Oliver Buchholz, and Florian Jug (2019). "Noise2void-learning denoising from single noisy images". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Nilesh Kulkarni, Abhinav Gupta, David F Fouhey, and Shubham Tulsiani (2020). "Articulation-aware canonical surface mapping". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Nilesh Kulkarni, Abhinav Gupta, and Shubham Tulsiani (2019). "Canonical surface mapping via geometric cycle consistency". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Tejas D Kulkarni, Ankush Gupta, Catalin Ionescu, Sebastian Borgeaud, Malcolm Reynolds, Andrew Zisserman, and Volodymyr Mnih (2019). "Unsupervised learning of object keypoints for perception and control". In: *Advances in neural information processing systems*.

Black Forest Labs (2023). *FLUX*. https://github.com/black-forest-labs/flux. URL: https://github.com/black-forest-labs/flux.

Jae Yong Lee, Joseph DeGol, Victor Fragoso, and Sudipta N Sinha (2021). "Patchmatch-based neighborhood consensus for semantic correspondence". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Junghyup Lee, Dohyung Kim, Jean Ponce, and Bumsub Ham (2019). "Sfnet: Learning object-aware semantic correspondence". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Jiabao Lei, Jiapeng Tang, and Kui Jia (2023). "RGBD2: Generative Scene Synthesis via Incremental View Inpainting Using RGBD Diffusion Models". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. (2022). "Solving quantitative reasoning problems with language models". In: *arXiv preprint arXiv:2206.14858*.

Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi (2024). "Instant3D: Fast Text-to-3D with Sparse-View Generation and Large Reconstruction Model". In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Jiaxin Li, Zijian Feng, Qi She, Henghui Ding, Changhu Wang, and Gim Hee Lee (2021). "MINE: Towards Continuous Depth MPI with NeRF for Novel View Synthesis". In: *Proceedings of the International Conference on Computer Vision (ICCV)*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi (2023). "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models". In: *International conference on machine learning*. PMLR.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi (2022). "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation". In: *International conference on machine learning*. PMLR.

Lei Li, Yuwei Yin, Shicheng Li, Liang Chen, Peiyi Wang, Shuhuai Ren, Mukai Li, Yazheng Yang, Jingjing Xu, Xu Sun, et al. (2023). "M^{3}IT: A Large-Scale Dataset towards Multi-Modal Multilingual Instruction Tuning". In: *arXiv preprint arXiv:2306.04387*.

Muchen Li and Leonid Sigal (2021). "Referring transformer: A one-step approach to multi-task visual grounding". In: *Advances in Neural Information Processing Systems*.

Shuda Li, Kai Han, Theo W Costain, Henry Howard-Jenkins, and Victor Prisacariu (2020). "Correspondence networks with adaptive neighbourhood consensus". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Wanhua Li, Renping Zhou, Jiawei Zhou, Yingwei Song, Johannes Herter, Minghan Qin, Gao Huang, and Hanspeter Pfister (2025). "4D LangSplat: 4D Language Gaussian Splatting via Multimodal Large Language Models". In: *arXiv preprint arXiv:2503.10437*.

Weiyu Li, Jiarui Liu, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long (2024). "CraftsMan: High-fidelity Mesh Generation with 3D Native Generation and Interactive Geometry Refiner". In: *arXiv*.

Xin Li, Deng-Ping Fan, Fan Yang, Ao Luo, Hong Cheng, and Zicheng Liu (2021). "Probabilistic model distillation for semantic correspondence". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Xinghui Li, Jingyi Lu, Kai Han, and Victor Prisacariu (2023). "SD4Match: Learning to Prompt Stable Diffusion Model for Semantic Matching". In: *arXiv preprint arXiv:2310.17569*.

Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan (2021). "Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm". In: *arXiv preprint arXiv:2110.05208*.

Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, and Yan-Pei Cao (2025). "TripoSG: High-Fidelity 3D Shape Synthesis using Large-Scale Rectified Flow Models". In: *arXiv*.

Zhengqi Li, Qianqian Wang, Noah Snavely, and Angjoo Kanazawa (2022). "Infinitenature-zero: Learning perpetual view generation of natural scenes from single images". In: *European Conference on Computer Vision*. Springer.

Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu (2022). "Open-vocabulary semantic segmentation with mask-adapted clip". In: *arXiv preprint arXiv:2210.04150*.

Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou (2022). "Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning". In: *Advances in Neural Information Processing Systems*.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin (2022). "Magic3D: High-Resolution Text-to-3D Content Creation". In: *arXiv.cs*.

Chieh Hubert Lin, Hsin-Ying Lee, Willi Menapace, Menglei Chai, Aliaksandr Siarohin, Ming-Hsuan Yang, and Sergey Tulyakov (2023). "Infinicity: Infinite-scale city synthesis". In: *Proceedings of the IEEE/CVF international conference on computer vision*.

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.

A Liu, R Tucker, V Jampani, A Makadia, and N Snavely... (2021). "Infinite Nature: Perpetual View Generation of Natural Scenes from a Single Image". In: *Proceedings of the International Conference on Computer Vision (ICCV)*.

Benlin Liu, Yuhao Dong, Yiqin Wang, Yongming Rao, Yansong Tang, Wei-Chiu Ma, and Ranjay Krishna (2024). "Coarse correspondence elicit 3d spacetime understanding in multimodal language model". In: *arXiv preprint arXiv:2408.00754*.

Daqing Liu, Hanwang Zhang, Feng Wu, and Zheng-Jun Zha (2019). "Learning to assemble neural module tree networks for visual grounding". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Dingning Liu, Xiaomeng Dong, Renrui Zhang, Xu Luo, Peng Gao, Xiaoshui Huang, Yongshun Gong, and Zhihui Wang (2023). "3daxiesprompts: Unleashing the 3d spatial task capabilities of gpt-4v". In: *arXiv preprint arXiv:2312.09738*.

Jingyu Liu, Liang Wang, and Ming-Hsuan Yang (2017). "Referring expression generation and comprehension via attributes". In: *Proceedings of the IEEE International Conference on Computer Vision*.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick (2023a). "Zero-1-to-3: Zero-shot One Image to 3D Object". In: *Proceedings of the International Conference on Computer Vision (ICCV)*.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick (2023b). "Zero-1-to-3: Zero-shot one image to 3d object". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

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*.

Yanbin Liu, Linchao Zhu, Makoto Yamada, and Yi Yang (2020). "Semantic correspondence as an optimal transport problem". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yongfei Liu, Bo Wan, Xiaodan Zhu, and Xuming He (2020a). "Learning cross-modal context graph for visual grounding". In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yongfei Liu, Bo Wan, Xiaodan Zhu, and Xuming He (2020b). "Learning cross-modal context graph for visual grounding". In: *Proceedings of the AAAI conference on artificial intelligence*.

Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell (2023a). "Diffusion Hyperfeatures: Searching Through Time and Space for Semantic Correspondence". In: *Advances in Neural Information Processing Systems*.

Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell (2023b). "Diffusion hyperfeatures: Searching through time and space for semantic correspondence". In: *Advances in Neural Information Processing Systems*.

Ruotian Luo and Gregory Shakhnarovich (2017). "Comprehension-guided referring expressions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy (2016). "Generation and comprehension of unambiguous object descriptions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Octave Mariotti, Oisin Mac Aodha, and Hakan Bilen (2024). "Improving semantic correspondence with viewpoint-guided spherical maps". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Joanna Materzyńska, Antonio Torralba, and David Bau (2022). "Disentangling visual and written concepts in CLIP". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, Natalia Neverova, Andrea Vedaldi, Oran Gafni, and Filippos Kokkinos (2024). "IM-3D: Iterative Multiview Diffusion and Reconstruction for High-Quality 3D Generation". In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, and Andrea Vedaldi (2023). "Realfusion: 360deg reconstruction of any object from a single image". In:

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

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

Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi (2022b). "Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Iaroslav Melekhov, Aleksei Tiulpin, Torsten Sattler, Marc Pollefeys, Esa Rahtu, and Juho Kannala (2019). "Dgc-net: Dense geometric correspondence network". In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE.

Quan Meng, Lei Li, Matthias Nießner, and Angela Dai (2024). "Lt3sd: Latent trees for 3d scene diffusion". In: *arXiv preprint arXiv:2409.08215*.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng (2021). "Nerf: Representing scenes as neural radiance fields for view synthesis". In: *Communications of the ACM*.

Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar (2019). "Local light field fusion: practical view synthesis with prescriptive sampling guidelines". In.

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: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Juhong Min and Minsu Cho (2021). "Convolutional hough matching networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho (2019). "Spair-71k: A large-scale benchmark for semantic correspondence". In: *arXiv preprint arXiv:1908.10543*.

Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho (2020). "Learning to compose hypercolumns for visual correspondence". In: *European Conference on Computer Vision*. Springer.

Luca Morreale, Noam Aigerman, Vladimir G Kim, and Niloy J Mitra (2024). "Neural semantic surface maps". In: *Computer Graphics Forum*. Wiley Online Library.

Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie (2022). "Slip: Self-supervision meets language-image pre-training". In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*. Springer.

Varun K Nagaraja, Vlad I Morariu, and Larry S Davis (2016). "Modeling context between objects for referring expression understanding". In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. Springer.

Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, et al. (2024). "Pivot: Iterative visual prompting elicits actionable knowledge for vlms". In: *arXiv preprint arXiv:2402.07872*.

Natalia Neverova, David Novotny, Marc Szafraniec, Vasil Khalidov, Patrick Labatut, and Andrea Vedaldi (2020). "Continuous surface embeddings". In: *Advances in Neural Information Processing Systems*.

Natalia Neverova, Artsiom Sanakoyeu, Patrick Labatut, David Novotny, and Andrea Vedaldi (2021). "Discovering relationships between object categories via universal canonical maps". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Alejandro Newell, Kaiyu Yang, and Jia Deng (2016). "Stacked Hourglass Networks for Human Pose Estimation". In: *ECCV*.

Thao Nguyen, Samir Yitzhak Gadre, Gabriel Ilharco, Sewoong Oh, and Ludwig Schmidt (2024). "Improving multimodal datasets with image captioning". In: *Advances in Neural Information Processing Systems*.

Dolev Ofri-Amar, Michal Geyer, Yoni Kasten, and Tali Dekel (2023). "Neural Congealing: Aligning Images to a Joint Semantic Atlas". In: *arXiv preprint arXiv:2302.03956*.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert (2017). "Feature Visualization". In: *Distill*. https://distill.pub/2017/feature-visualization.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals (2018). "Representation learning with contrastive predictive coding". In: *arXiv preprint arXiv:1807.03748*.

OpenAI (2023). *GPT-4: Generative Pre-trained Transformer 4*. https://openai.com/. Accessed: yyyy-mm-dd.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. (2023). "Dinov2: Learning robust visual features without supervision". In: *arXiv preprint arXiv:2304.07193*.

Hao Ouyang, Kathryn Heal, Stephen Lombardi, and Tiancheng Sun (2023). "Text2immersion: Generative immersive scene with 3d gaussians". In: *arXiv preprint arXiv:2312.09242*.

Daniil Pakhomov, Sanchit Hira, Narayani Wagle, Kemar E Green, and Nassir Navab (2021). "Segmentation in style: Unsupervised semantic image segmentation with stylegan and CLIP". In: *arXiv preprint arXiv:2107.12518*.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami (2016). "The limitations of deep learning in adversarial settings". In: *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho (2019). "Relational knowledge distillation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

William Peebles, Jun-Yan Zhu, Richard Zhang, Antonio Torralba, Alexei A Efros, and Eli Shechtman (2022). "Gan-supervised dense visual alignment". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

T.D. Pereira, D. E. Aldarondo, L. Willmore, M. Kislin, S. S.-H. Wang, M. Murthy, and J. W. Shaevitz (2018). "Fast animal pose estimation using deep neural networks". In: *bioRxiv*.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall (2023). "DreamFusion: Text-to-3D using 2D Diffusion". In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Simon Prince, Peng Li, Yun Fu, Umar Mohammed, and James H. Elder (2012). "Probabilistic Models for Inference about Identity". In.

Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, and Bernard Ghanem (2023). "Magic123: One Image to High-Quality 3D Object Generation Using Both 2D and 3D Diffusion Priors". In: *arXiv.cs*.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. (2019). "Language models are unsupervised multitask learners". In: *OpenAI blog*.

Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Nataniel Ruiz, Ben Mildenhall, Shiran Zada, Kfir Aberman, Michael Rubinstein, Jonathan Barron, Yuanzhen Li, and Varun Jampani (2023). "DreamBooth3D: Subject-Driven Text-to-3D Generation". In: *Proceedings of the International Conference on Computer Vision (ICCV)*.

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

Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas (2021). "Humor: 3d human motion model for robust pose estimation". In: *Proceedings of the IEEE/CVF international conference on computer vision*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun (2015). "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*.

Razieh Rezaei, Masoud Jalili Sabet, Jindong Gu, Daniel Rueckert, Philip Torr, and Ashkan Khakzar (2024). "Learning Visual Prompts for Guiding the Attention of Vision Transformers". In: *arXiv preprint arXiv:2406.03303*.

I. Rocco, R. Arandjelović, and J. Sivic (2017). "Convolutional neural network architecture for geometric matching". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

I. Rocco, R. Arandjelović, and J. Sivic (n.d.). "End-to-end weakly-supervised semantic alignment". In: *arXiv preprint arXiv:1712.06861* ().

I. Rocco, M. Cimpoi, R. Arandjelović, A. Torii, T. Pajdla, and J. Sivic (2018). "Neighbourhood Consensus Networks". In: *Proceedings of the 32nd Conference on Neural Information Processing Systems*.

Ignacio Rocco, Relja Arandjelovic, and Josef Sivic (2017). "Convolutional neural network architecture for geometric matching". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Ignacio Rocco, Relja Arandjelović, and Josef Sivic (2018). "End-to-end weakly-supervised semantic alignment". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Ignacio Rocco, Relja Arandjelović, and Josef Sivic (2020). "Efficient neighbourhood consensus networks via submanifold sparse convolutions". In: *European conference on computer vision*. Springer.

Chris Rockwell, David F. Fouhey, and Justin Johnson (2021). "PixelSynth: Generating a 3D-Consistent Experience from a Single Image". In: *Proceedings of the International Conference on Computer Vision (ICCV)*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer (2021). *High-Resolution Image Synthesis with Latent Diffusion Models*. arXiv: 2112.10752 [cs.CV].

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer (2022). "High-Resolution Image Synthesis with Latent Diffusion Models". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

Serim Ryou and Pietro Perona (2021). "Weakly supervised keypoint discovery". In: *arXiv preprint arXiv:2109.13423*.

Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, and Jiajun Wu (2023). "ZeroNVS: Zero-Shot 360-Degree View Synthesis from a Single Real Image". In: *arXiv.cs*.

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*.

Christian Schlarmann and Matthias Hein (2023). "On the adversarial robustness of multi-modal foundation models". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. (2022). "Laion-5b: An open large-scale dataset for training next generation image-text models". In: *Advances in Neural Information Processing Systems*.

Junyoung Seo, Kazumi Fukuda, Takashi Shibuya, Takuya Narihira, Naoki Murata, Shoukang Hu, Chieh-Hsin Lai, Seungryong Kim, and Yuki Mitsufuji (2024). "GenWarp: Single Image to Novel Views with Semantic-Preserving Generative Warping". In: *arXiv preprint arXiv:2405.17251*.

Paul Hongsuck Seo, Jongmin Lee, Deunsol Jung, Bohyung Han, and Minsu Cho (2018). "Attentive semantic alignment with offset-aware correlation kernels". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Yu Shang, Yuming Lin, Yu Zheng, Hangyu Fan, Jingtao Ding, Jie Feng, Jiansheng Chen, Li Tian, and Yong Li (2024). "UrbanWorld: An Urban World Model for 3D City Generation". In: *arXiv preprint arXiv:2407.11965*.

Sheng Shen, Shijia Yang, Tianjun Zhang, Bohan Zhai, Joseph E Gonzalez, Kurt Keutzer, and Trevor Darrell (2022). "Multitask Vision-Language Prompt Tuning". In: *arXiv preprint arXiv:2211.11720*.

Hengcan Shi, Munawar Hayat, Yicheng Wu, and Jianfei Cai (2022). "ProposalCLIP: Unsupervised Open-Category Object Proposal Generation via Exploiting CLIP Cues". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang (2024). "MVDream: Multi-view Diffusion for 3D Generation". In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang (2020). "3D Photography Using Context-Aware Layered Depth Inpainting". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jaidev Shriram, Alex Trevithick, Lingjie Liu, and Ravi Ramamoorthi (2025). "RealmDreamer: Text-Driven 3D Scene Generation with Inpainting and Depth Diffusion". In: *Proceedings of the International Conference on 3D Vision (3DV)*.

Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi (2023a). "Learning Universal Semantic Correspondences with No Supervision and Automatic Data Curation". In: *Proc. IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*.

Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi (2023b). "What does clip know about a red circle? visual prompt engineering for vlms". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi (2024). "SHIC: Shape-Image Correspondences with no Keypoint Supervision". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Richard Sinkhorn and Paul Knopp (1967). “Concerning nonnegative matrices and doubly stochastic matrices”. In: *Pacific Journal of Mathematics*.

Daeun Song, Jing Liang, Xuesu Xiao, and Dinesh Manocha (2024). “TGS: Trajectory Generation and Selection using Vision Language Models in Mapless Outdoor Environments”. In: *arXiv preprint arXiv:2408.02454*.

Dan Song, Xinwei Fu, Ning Liu, Weizhi Nie, Wenhui Li, Lanjun Wang, You Yang, and An-An Liu (2025). “Mv-clip: Multi-view clip for zero-shot 3d shape recognition”. In: *IEEE Transactions on Circuits and Systems for Video Technology*.

Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely (2019). “Pushing the boundaries of view extrapolation with multiplane images”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Gabriela Ben Melech Stan, Diana Wofk, Scottie Fox, Alex Redden, Will Saxton, Jean Yu, Estelle Aflalo, Shao-Yen Tseng, Fabio Nonato, Matthias Muller, and Vasudev Lal (2023). “LDM3D: Latent Diffusion Model for 3D”. In: *arXiv.cs*.

Kenneth O Stanley (2007). “Compositional pattern producing networks: A novel abstraction of development”. In: *Genetic programming and evolvable machines*.

Sanjay Subramanian, William Merrill, Trevor Darrell, Matt Gardner, Sameer Singh, and Anna Rohrbach (2022). “Reclip: A strong zero-shot baseline for referring expression comprehension”. In: *arXiv preprint arXiv:2204.05991*.

Quan Sun, Jinsheng Wang, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, and Xinlong Wang (2024). “Eva-clip-18b: Scaling clip to 18 billion parameters”. In: *arXiv preprint arXiv:2402.04252*.

Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi (2023). “Viewset Diffusion: (0-)Image-Conditioned 3D Generative Models from 2D Data”. In: *Proceedings of the International Conference on Computer Vision (ICCV)*.

Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu (2024). “LGM: Large Multi-View Gaussian Model for High-Resolution 3D Content Creation”. In: *arXiv preprint*.

Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan (2023). “Emergent Correspondence from Image Diffusion”. In: *Thirty-seventh Conference on Neural Information Processing Systems*.

James Thewlis, Samuel Albanie, Hakan Bilen, and Andrea Vedaldi (2019). “Unsupervised learning of landmarks by descriptor vector exchange”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

James Thewlis, Hakan Bilen, and Andrea Vedaldi (2017a). "Unsupervised learning of object frames by dense equivariant image labelling". In: *NeurIPS*.

James Thewlis, Hakan Bilen, and Andrea Vedaldi (2017b). "Unsupervised learning of object frames by dense equivariant image labelling". In: *Advances in neural information processing systems*.

James Thewlis, Hakan Bilen, and Andrea Vedaldi (2018). "Modelling and unsupervised learning of symmetric deformable object categories". In: *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*.

Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li (2016). "Yfcc100m: The new data in multimedia research". In: *Communications of the ACM*.

Junjiao Tian, Lavisha Aggarwal, Andrea Colaco, Zsolt Kira, and Mar Gonzalez-Franco (2024). "Diffuse attend and segment: Unsupervised zero-shot segmentation using stable diffusion". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Prune Truong, Martin Danelljan, Luc V Gool, and Radu Timofte (2020a). "GOCor: Bringing globally optimized correspondence volumes into your neural network". In: *Advances in Neural Information Processing Systems*.

Prune Truong, Martin Danelljan, and Radu Timofte (2020b). "GLU-Net: Global-local universal network for dense flow and correspondences". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Prune Truong, Martin Danelljan, Fisher Yu, and Luc Van Gool (2021). "Warp consistency for unsupervised learning of dense correspondences". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Prune Truong, Martin Danelljan, Fisher Yu, and Luc Van Gool (2022). "Probabilistic Warp Consistency for Weakly-Supervised Semantic Correspondences". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. (2025). "SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features". In: *arXiv preprint arXiv:2502.14786*.

Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi (2022). "Neural feature fusion fields: 3d distillation of self-supervised 2d image representations". In: *arXiv preprint arXiv:2209.03494*.

Cheng-Hao Tu, Zheda Mai, and Wei-Lun Chao (2022). “Visual Query Tuning: Towards Effective Usage of Intermediate Representations for Parameter and Memory Efficient Transfer Learning”. In: *arXiv preprint arXiv:2212.03220*.

Richard Tucker and Noah Snavely (2020). “Single-view view synthesis with multiplane images”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Shubham Tulsiani, Richard Tucker, and Noah Snavely (2018). “Layer-Structured 3D Scene Inference via View Synthesis”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel (2022a). “Splicing ViT Features for Semantic Appearance Transfer”. In.

Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel (2022b). “Splicing ViT Features for Semantic Appearance Transfer”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Narek Tumanyan, Assaf Singer, Shai Bagon, and Tali Dekel (2024). “Dino-tracker: Taming dino for self-supervised point tracking in a single video”. In: *European Conference on Computer Vision*. Springer.

Nikolai Ufer and Björn Ommer (2017). “Deep Semantic Feature Matching”. In.

Lukas Uzolas, Elmar Eisemann, and Petr Kellnhofer (2025). “Surface-Aware Distilled 3D Semantic Features”. In: *arXiv preprint arXiv:2503.18254*.

Yael Vinker, Ehsan Pajouheshgar, Jessica Y Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir (2022). “Clipasso: Semantically-aware object sketching”. In: *ACM Transactions on Graphics (TOG)*.

Urs Waldmann, Alex Hoi Hang Chan, Hemal Naik, Máté Nagy, Iain D Couzin, Oliver Deussen, Bastian Goldluecke, and Fumihiro Kano (2023). “3d-muppet: 3d multi-pigeon pose estimation and tracking”. In: *arXiv preprint arXiv:2308.15316*.

Guangcong Wang, Peng Wang, Zhaoxi Chen, Wenping Wang, Chen Change Loy, and Ziwei Liu (2024). “PERF: Panoramic Neural Radiance Field from a Single Panorama”. In: *tpami*.

Jialu Wang, Yang Liu, and Xin Eric Wang (2021). “Are gender-neutral queries really gender-neutral? mitigating gender bias in image search”. In: *arXiv preprint arXiv:2109.05433*.

Liwei Wang, Yin Li, and Svetlana Lazebnik (2016). “Learning deep structure-preserving image-text embeddings”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel (2019a). "Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel (2019b). "Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud (2024). "DUSt3R: Geometric 3D Vision Made Easy". In: *CVPR*.

Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang (2023). "Images speak in images: A generalist painter for in-context visual learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra (2023). "Cut and learn for unsupervised object detection and instance segmentation". In: *arXiv preprint arXiv:2301.11320*.

Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L Crowley, and Dominique Vaufreydaz (2022a). "Self-supervised transformers for unsupervised object discovery using normalized cut". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L Crowley, and Dominique Vaufreydaz (2022b). "TokenCut: Segmenting Objects in Images and Videos with Self-supervised Transformer and Normalized Cut". In: *arXiv preprint arXiv:2209.00383*.

Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu (2022). "Cris: Clip-driven referring image segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu (2023). "Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation". In: *Advances in Neural Information Processing Systems*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. (2022). "Chain-of-thought prompting elicits

reasoning in large language models". In: *Advances in neural information processing systems*.

Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh (2016). "Convolutional Pose Machines". In: *CVPR*.

Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona (2010). "Caltech-UCSD birds 200". In.

Licheng Wen, Xuemeng Yang, Daocheng Fu, Xiaofeng Wang, Pinlong Cai, Xin Li, Tao Ma, Yingxuan Li, Linran Xu, Dengke Shang, et al. (2023). "On the road with gpt-4v (ision): Early explorations of visual-language model on autonomous driving". In: *arXiv preprint arXiv:2311.05332*.

Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson (2020). "SynSin: End-to-End View Synthesis From a Single Image". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

World Labs (2024). *Generating Worlds*. URL: https://www.worldlabs.ai/blog.

Jiannan Wu, Muyan Zhong, Sen Xing, Zeqiang Lai, Zhaoyang Liu, Zhe Chen, Wenhai Wang, Xizhou Zhu, Lewei Lu, Tong Lu, et al. (2024). "Visionllm v2: An end-to-end generalist multimodal large language model for hundreds of vision-language tasks". In: *Advances in Neural Information Processing Systems*.

Junda Wu, Zhehao Zhang, Yu Xia, Xintong Li, Zhaoyang Xia, Aaron Chang, Tong Yu, Sungchul Kim, Ryan A Rossi, Ruiyi Zhang, et al. (2024). "Visual Prompting in Multimodal Large Language Models: A Survey". In: *arXiv preprint arXiv:2409.15310*.

Shangzhe Wu, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi (2021). "DOVE: Learning Deformable 3D Objects by Watching Videos". In.

Shangzhe Wu, Ruining Li, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi (2023). "MagicPony: Learning Articulated 3D Animals in the Wild". In: *CVPR*.

Zhennan Wu, Yang Li, Han Yan, Taizhang Shang, Weixuan Sun, Senbo Wang, Ruikai Cui, Weizhe Liu, Hiroyuki Sato, Hongdong Li, et al. (2024). "Blockfusion: Expandable 3d scene generation using latent tri-plane extrapolation". In: *ACM Transactions on Graphics (TOG)*.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang (2024a). "Structured 3D Latents for Scalable and Versatile 3D Generation". In: *arXiv preprint*.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang (2024b). "Structured 3d latents for scalable and versatile 3d generation". In: *arXiv preprint arXiv:2412.01506*.

Jianfeng Xiang, Jiaolong Yang, Binbin Huang, and Xin Tong (2023). "3d-aware image generation using 2d diffusion models". In: *Proceedings of the International Conference on Computer Vision (ICCV)*.

Haozhe Xie, Zhaoxi Chen, Fangzhou Hong, and Ziwei Liu (2024). "Citydreamer: Compositional generative model of unbounded 3d cities". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Jinheng Xie, Songhe Deng, Bing Li, Haozhe Liu, Yawen Huang, Yefeng Zheng, Jurgen Schmidhuber, Bernard Ghanem, Linlin Shen, and Mike Zheng Shou (2024). "Tune-An-Ellipse: CLIP Has Potential to Find What You Want". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Junyu Xie, Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman (2024). "AutoAD-Zero: A Training-Free Framework for Zero-Shot Audio Description". In: *arXiv preprint arXiv:2407.15850*.

Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang (2022). "Groupvit: Semantic segmentation emerges from text supervision". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein (2024). "GRM: Large Gaussian Reconstruction Model for Efficient 3D Reconstruction and Generation". In: *arXiv preprint*.

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: *arXiv:2401.10891*.

Lingfeng Yang, Yueze Wang, Xiang Li, Xinlong Wang, and Jian Yang (2023). "Fine-grained visual prompting". In: *Advances in Neural Information Processing Systems*.

Lingfeng Yang, Yueze Wang, Xiang Li, Xinlong Wang, and Jian Yang (2024). "Fine-grained visual prompting". In: *Advances in Neural Information Processing Systems*.

Sibei Yang, Guanbin Li, and Yizhou Yu (2019a). "Dynamic graph attention for referring expression comprehension". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Sibei Yang, Guanbin Li, and Yizhou Yu (2019b). "Dynamic graph attention for referring expression comprehension". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang (2021). "Crossing the format boundary of text and boxes: Towards unified vision-language modeling". In: *arXiv preprint arXiv:2111.12085*.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang (2023). "The dawn of lmms: Preliminary explorations with gpt-4v (ision)". In: *arXiv preprint arXiv:2309.17421*.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. (2024). "CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer". In: *arXiv preprint arXiv:2408.06072*.

Zongming Yang, Liang Yang, Liren Kong, Ailin Wei, Jesse Leaman, Johnell Brooks, and Bing Li (2022). "Seeway: Vision-language assistive navigation for the visually impaired". In: *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE.

Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu (2021). "FILIP: fine-grained interactive language-image pre-training". In: *arXiv preprint arXiv:2111.07783*.

Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun (2021). "Cpt: Colorful prompt tuning for pre-trained vision-language models". In: *arXiv preprint arXiv:2109.11797*.

Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun (2024). "Cpt: Colorful prompt tuning for pre-trained vision-language models". In: *AI Open*.

Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim (2017). "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T Freeman, and Jiajun Wu (2024a). "Wonderworld: Interactive 3d scene generation from a single image". In: *arXiv preprint arXiv:2406.09394*.

Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T Freeman, Forrester Cole, Deqing Sun, Noah Snavely, Jiajun Wu, et al. (2024b). "Wonderjourney: Going from anywhere to everywhere". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg (2018). “Mattnet: Modular attention network for referring expression comprehension”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg (2016). “Modeling context in referring expressions”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer.

Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy (2022). “Unified vision and language prompt learning”. In: *arXiv preprint arXiv:2210.07225*.

Jana Zeller, Aleksandar Shtedritski, and Christian Rupprecht (2025). “Highlight: Learning Visual Prompts for Vision-Language Models”. In.

Guanqi Zhan, Chuanxia Zheng, Weidi Xie, and Andrew Zisserman (2024). “A general protocol to probe large vision models for 3d physical understanding”. In: *Advances in Neural Information Processing Systems*.

Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka (2023). “3DShape2VecSet: A 3D Shape Representation for Neural Fields and Generative Diffusion Models”. In: *ACM Transactions on Graphics*.

Hongwen Zhang, Yating Tian, Xinchi Zhou, Wanli Ouyang, Yebin Liu, Limin Wang, and Zhenan Sun (2021). “Pymaf: 3d human pose and shape regression with pyramidal mesh alignment feedback loop”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Jingbo Zhang, Xiaoyu Li, Ziyu Wan, Can Wang, and Jing Liao (2024). “Text2nerf: Text-driven 3d scene generation with neural radiance fields”. In: *IEEE Transactions on Visualization and Computer Graphics*.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang (2023a). “A Tale of Two Features: Stable Diffusion Complements DINO for Zero-Shot Semantic Correspondence”. In: *arXiv preprint arxiv:2305.15347*.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Eric Chen, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang (2023b). “Telling Left from Right: Identifying Geometry-Aware Semantic Correspondence”. In: *arXiv.cs*.

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu (2024). “CLAY: A Controllable Large-scale

Generative Model for Creating High-quality 3D Assets". In: *ACM Transactions on Graphics (TOG)*.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala (2023). *Adding Conditional Control to Text-to-Image Diffusion Models*.

Ning Zhang, Jeff Donahue, Ross B. Girshick, and Trevor Darrell (2014). "Part-Based R-CNNs for Fine-Grained Category Detection". In: *ECCV*.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao (2021). "Vinvl: Revisiting visual representations in vision-language models". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li (2022). "Pointclip: Point cloud understanding by clip". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang (2018). "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sheng Zhang, Salman Khan, Zhiqiang Shen, Muzammal Naseer, Guangyi Chen, and Fahad Khan (2022). "PromptCAL: Contrastive Affinity Learning via Auxiliary Prompts for Generalized Novel Category Discovery". In: *arXiv preprint arXiv:2212.05590*.

Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee (2018). "Unsupervised discovery of object landmarks as structural representations". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Dongyang Zhao, Ziyang Song, Zhenghao Ji, Gangming Zhao, Weifeng Ge, and Yizhou Yu (2021). "Multi-scale Matching Networks for Semantic Correspondence". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. (2025). "Hunyuan3D 2.0: Scaling Diffusion Models for High Resolution Textured 3D Assets Generation". In: *arXiv preprint arXiv:2501.12202*.

Feng Zhou, Xinyu Zhang, Hui Shuai, Renlong Hang, Shanshan Zhu, and Tianyu Geng (2025). "Dual-Granularity Feature Alignment for Change Detection in Remote Sensing Images". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu (2022a). “Conditional prompt learning for vision-language models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu (2022b). “Learning to prompt for vision-language models”. In: *International Journal of Computer Vision*.

Tinghui Zhou, Philipp Krahenbuhl, Mathieu Aubry, Qixing Huang, and Alexei A Efros (2016). “Learning dense correspondence via 3d-guided cycle consistency”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Silvia Zuffi, Angjoo Kanazawa, Tanya Berger-Wolf, and Michael J Black (2019). “Three-D Safari: Learning to Estimate Zebra Pose, Shape, and Texture from Images" In the Wild"”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Silvia Zuffi, Angjoo Kanazawa, and Michael J. Black (2018). “Lions and Tigers and Bears: Capturing Non-rigid, 3D, Articulated Shape from Images”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Silvia Zuffi, Angjoo Kanazawa, David W. Jacobs, and Michael J. Black (2017). “3D Menagerie: Modeling the 3D Shape and Pose of Animals”. In: *CVPR*.

Statement of Authorship for the paper "Learning Universal Semantic Correspondences With No Supervision and Automatic Data Curation" in Chapter 2.

Paper title	Learning Universal Semantic Correspondences With No Supervision and Automatic Data Curation
Authors	Aleksandar Shtedritski, Andrea Vedaldi, Christian Rupprecht
Publication status	Published
Publication details	International Conference on Computer Vision (ICCV) LIMIT Wokrshop, 2023.

Student Confirmation

Student name	Aleksandar Shtedritski	
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	Aleksandar Shtedritski	June 28th 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	Prof. Christian Rupprecht	
Supervisor comments		
Signature and Date		29/06/2025

Statement of Authorship for the paper "SHIC: Shape-Image Correspondences with No Keypoint Supervision" in Chapter 3.

Paper title	SHIC: Shape-Image Correspondences with No Keypoint Supervision
Authors	Aleksandar Shtedritski, Christian Rupprecht, Andrea Vedaldi
Publication status	Published
Publication details	European Conference on Computer Vision (ECCV), 2024.

Student Confirmation

Student name	Aleksandar Shtedritski	
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	Aleksandar Shtedritski	June 28th 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	Prof. Christian Rupprecht	
Supervisor comments		
Signature and Date		29/06/2025

Statement of Authorship for the paper "What Does CLIP Know About a Red Circle? Visual Prompt Engineering for VLMs" in Chapter 4.

Paper title	What Does CLIP Know About a Red Circle? Visual Prompt Engineering for VLMs
Authors	Aleksandar Shtedritski, Christian Rupprecht, Andrea Vedaldi
Publication status	Published
Publication details	International Conference on Computer Vision (ICCV), 2023.

Student Confirmation

Student name	Aleksandar Shtedritski	
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	Aleksandar Shtedritski	June 28th 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	Prof. Christian Rupprecht	
Supervisor comments		
Signature and Date		29/06/2025

Statement of Authorship for the paper "Highlight: Learning Visual Prompts for Vision-Language Models" in Chapter 5.

Paper title	Highlight: Learning Visual Prompts for Vision-Language Models
Authors	Jana Zeller, Aleksandar Shtedritski, Christian Rupprecht
Publication status	Published
Publication details	Computer Vision and Pattern Recognition Conference (CVPR) Eval-FoMo Workshop, 2025.

Student Confirmation

Student name	Aleksandar Shtedritski	
Contribution to the paper	Second author contribution: • conception of research ideas • proof of concept implementation of models • writing and presentation of the paper	
Signature and Date	Aleksandar Shtedritski	June 28th 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	Prof. Christian Rupprecht	
Supervisor comments		
Signature and Date		29/06/2025

Statement of Authorship for the paper "SynCity: Training-Free Generation of 3D Worlds" in Chapter 6.

Paper title	SynCity: Training-Free Generation of 3D Worlds
Authors	Paul Engslter*, Aleksandar Shtedritski*, Iro Laina, Christian Rupprecht, Andrea Vedaldi
Publication status	Accepted for publication
Publication details	International Conference on Computer Vision (ICCV), 2025.

Student Confirmation

Student name	Aleksandar Shtedritski	
Contribution to the paper	Co-first author contribution: • joint conception of research ideas • implementation of inpainting and blending pipelines • writing and presentation of the paper	
Signature and Date	Aleksandar Shtedritski	June 28th 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	Prof. Christian Rupprecht	
Supervisor comments		
Signature and Date		29/06/2025