

Imperial College of Science, Technology and Medicine
Department of Computing

Scalable 3D Generative Reconstruction for Spatial AI

Xin Kong

Mar 2026

Supervised by Prof. Andrew J. Davison

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy of Imperial College London.

Statement of Originality

I declare that this thesis is written by myself and the work presented here is my own, unless stated otherwise.

Copyright Declaration

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Abstract

Spatial Artificial Intelligence (Spatial AI) aims to equip intelligent systems, such as robots, autonomous vehicles, and AR/VR devices, with the ability to perceive, map, and reason about the physical world in a structured and predictive manner. Achieving this requires 3D representations that are not only geometrically accurate but also scalable, compositional, and generative, enabling systems to reconstruct environments and synthesise consistent novel views. While classical SLAM systems and neural scene representations such as NeRF have achieved strong results in localisation and rendering, they remain limited by per-scene optimisation, limited scalability, and poor cross-scene generalisation. Meanwhile, recent advances in diffusion and autoregressive generative models have transformed image and video synthesis, yet extending these approaches to 3D and multi-view generation introduces new challenges in efficiency, consistency, and structured dependency modelling.

This thesis advances Spatial AI through three contributions addressing these challenges. First, we introduce *vMAP*, a vectorised object-level neural mapping framework that represents scenes as sets of learnable primitives. By combining object-centric decomposition with parallel optimisation, *vMAP* enables scalable and compositional neural mapping. Second, we propose *EscherNet*, a diffusion-based generative model for scalable multi-view synthesis. By learning cross-scene priors and decoupling generation from explicit 3D coordinates, *EscherNet* generalises to unseen environments and supports flexible conditioning on arbitrary input views. Finally, we present *CausNVS*, an autoregressive multi-view diffusion framework that introduces causal masking to model structured dependencies between reference and target views, enabling consistent and flexible view synthesis.

Together, these contributions advance Spatial AI toward scalable and generative 3D world modelling, demonstrating how structured mapping, diffusion-based synthesis, and autoregressive conditioning can be unified for predictive spatial understanding.

Acknowledgements

I feel so fortunate and honored to have pursued my PhD in this renowned lab, and I would like to express my sincere gratitude to my supervisor, Andy. His guidance and encouragement, together with his critical thinking and open attitude, allowed me to explore different techniques and later connect them back to Spatial AI. His philosophy of seeking “blue LED” research problems, and his emphasis that demos speak louder than table numbers, have had a profound impact on shaping my research, far beyond knowledge and publications alone. I am also truly thankful for the lab atmosphere: the daily lunch conversations on gaming, sports, history, news, books, and movies; the conference gatherings and trips; the funny jokes and wild ideas over pints; and the karaoke nights and party dancing. These moments will remain among my most precious memories of these four years.

Thanks to my labmates for the countless coffee breaks, pints, discussions, and laughter that made our time together so meaningful. Their support in research, technical challenges, and everyday encouragement has been invaluable. Special thanks to Marwan, always supportive and a reliable problem solver; to Shikun, who brought fresh machine learning perspectives that sparked new ideas; and to Hiden, whose encouragement and late-night companionship in the lab gave me strength to keep moving forward. I am also grateful to Eric, Riku, Ivan, Kirill, and Aalok for their insights into multi-view geometry, optimisation, robot learning, world culture, British knowledge, and all sorts of unexpected trivia. I have benefitted greatly from senior members including Edgar, Binbin, Ignacio, Joe, Callum, Seth, Gwangbin, Dorian, Tristan, and Kentaro, and I also thank Yifei, Nick, Yuzhou, Shinjeong, and Anagh. A heartfelt thanks goes to Iosifina, whose help with all the paperwork allowed us to focus fully on research.

Thanks also to my old friends in this field, Guangyao, Xiaoyang, and Xingguang, who have always been there for sharing thoughts, doubts, and encouragement.

Finally, my deepest gratitude and love go to my mom and dad. Their open-mindedness, boundless love, and unwavering support gave me warmth and the courage to explore the world without fear. Thanks to my partner Dr.Yicheng, meeting her has been the greatest blessing of this journey. Her strength in difficult times and her joy in moments of celebration have brightened every day. I feel truly fortunate to have shared both this PhD and our travels around the world with her, and I look forward with all my heart to every step of our future together.

Contents

1	Introduction	1
1.1	Spatial AI	1
1.2	3D Representations	4
1.3	Generative Models	6
1.4	Towards World Modelling	8
1.5	Contributions	9
1.6	Publications	10
1.7	Thesis Structure	11
2	Preliminaries	12
2.1	Notation	13
2.2	Camera Model	14
2.3	Advanced 3D Representations	15
2.4	Visual SLAM	17
2.5	Transformers and Attention	21
2.6	Diffusion Models	23

2.6.1	Discrete Formulation (DDPM)	23
2.6.2	Continuous Formulation (SDE)	25
2.6.3	Conditioning and Guidance in Practice	25
2.6.4	Recent Formulations	26
2.6.5	Key Advanced Applications	27
2.7	Autoregressive Models	29
2.7.1	Attention Mask	29
2.7.2	Key Advanced Applications	31
2.7.3	Comparing Autoregressive and Diffusion Models	33
2.7.4	Towards World Models	34
3	vMAP: Vectorised Object-level Neural Mapping	37
3.1	Introduction	38
3.2	Related Work	39
3.3	vMAP: Object Mapping System with Vectorised Training	41
3.3.1	System Overview	41
3.3.2	Vectorised Object Level Mapping	41
3.3.3	Neural Implicit Mapping	44
3.3.4	Compositional Scene Rendering	46
3.4	Experiments	46
3.4.1	Experimental Setup	46
3.4.2	Evaluation on Scene and Object Reconstruction	49

3.4.3	Performance Analysis	56
3.4.4	Real-to-Sim-to-Real for Robot Learning	59
3.5	Conclusion	60
4	EscherNet: Scalable 3D Multi-view Generative Synthesis	62
4.1	Introduction	64
4.2	Related Work	66
4.3	EscherNet	68
4.3.1	Architecture Design	70
4.3.2	Camera Positional Encoding (CaPE)	71
4.4	Experiments	76
4.4.1	Implementation Details	76
4.4.2	Evaluation Details	77
4.4.3	Results on Novel View Synthesis	78
4.4.4	Results on 3D Generation	81
4.4.5	Results on Text-to-3D Generation	83
4.4.6	Results with 6 DoF CaPE	85
4.5	Discussions	87
4.6	Conclusions	89
5	CausNVS: Autoregressive Multi-view Diffusion for 3D View Synthesis	91
5.1	Introduction	93

5.2	Related Work	95
5.3	CausNVS: Causal Diffusion Model for Novel View Synthesis	97
5.3.1	Training Objective	97
5.3.2	Random Per-frame Noise Conditioning	99
5.3.3	Relative Camera Pose Encoding (CaPE)	99
5.3.4	KV Caching as Spatial Memory	101
5.4	Experiments	102
5.4.1	Novel View Synthesis Results	104
5.4.2	Autoregressive v.s. Non-autoregressive	107
5.5	Conclusion	109
6	Conclusions and Future Work	112
6.1	Summary of Contributions	112
6.2	Future Work	113
	Bibliography	115

List of Tables

3.1	Overview of Evaluated Datasets	47
3.2	Reconstruction Results for 8 Indoor Replica Scenes	47
3.3	Scene-level Reconstruction Results on 8 Indoor Replica Scenes	50
3.4	Object-level Reconstruction Results on 8 Indoor Replica Scenes	51
3.5	2D Novel View Synthesis Rendering Results on Replica Dataset	51
3.6	Camera Tracking Results on TUM RGB-D Dataset	53
3.7	Memory and Runtime Efficiency	57
4.1	Novel View Synthesis Results on GSO and RTMV Datasets	78
4.2	Novel View Synthesis Results on NeRF Synthetic Dataset	79
4.3	Novel View Synthesis on NeRF Synthetic Dataset	80
4.4	3D Reconstruction Results on GSO	81
4.5	Additional Text-to-3D Generation with SDXL and MVDream	85
4.6	6DoF v.s. 4DoF CaPE	86
5.1	Novel View Synthesis Benchmark	102

5.2 **Novel view synthesis metrics on Re10K and LLFF.** Note: SEVA results on Re10K are obtained by running the official public code. No scale sweeping is applied at $N=1$. 105

5.3 N-to-M Novel View Synthesis Ablation 107

List of Figures

1.1	Applications of Spatial AI and overview of the thesis contributions	2
2.1	Illustration of the Pinhole Camera Model	14
2.2	Illustration of Neural Radiance Fields (NeRF)	15
2.3	Illustration of 3D Gaussian Splatting	15
2.4	Visual SLAM History	17
2.5	Towards Transformer based SLAM	20
2.6	The Transformer Architecture	21
2.7	Illustration of Attention	22
2.8	Diffusion DDPM Formulation	24
2.9	Diffusion SDE Formulation	25
2.10	Latent Diffusion Model	27
2.11	Align Your Latents Video Diffusion Model	28
2.12	Zero-1-to-3 View Synthesis Diffusion Model	29
2.13	Autoregressive Sequence Modeling	30
2.14	Different Attention Mask Patterns	30

2.15 Three Types of Visual Autoregressive Models	31
2.16 Teacher Forcing, Diffusion Forcing, and Self-Forcing	33
2.17 Conceptual Role of World Models	34
2.18 Goals and Approaches of World Models	35
3.1 Overview of vMAP Reconstruction Results	38
3.2 Overview of Training and Rendering Pipeline of vMAP	42
3.3 Visualisation of Depth-guided Sampling	45
3.4 Scene Reconstruction Visualisation for 4 Selected Replica Scenes	48
3.5 Visual Comparison at Object-level Reconstruction	49
3.6 Reconstruction Visualisation on ScanNet Dataset	49
3.7 Visualisation of 2D Novel View Synthesis on Replica Dataset	52
3.8 Visualisation of Reconstruction on TUM RGB-D Dataset	53
3.9 Real-world Live Reconstruction (Table-top)	54
3.10 Real-world Live Reconstruction (Scene-level)	54
3.11 Visualisation of Object Reconstructions Trained with Different Scales	55
3.12 Visualisation of Object-level Hole-filling	56
3.13 Vectorised v.s. Sequential Training	57
3.14 Object-level Reconstruction Performance vs. Model Size	58
3.15 Load vMAP Reconstruction in Issac Gym Robotic Simulation	60
4.1 Multi-view to Multi Novel View Synthesis	63

4.2	3D Representations Overview	67
4.3	EscherNet Architecture	69
4.4	Novel View Synthesis Visualisation on GSO and RTMV Datasets	82
4.5	Single View 3D Reconstruction Visualisation on GSO	82
4.6	Text-to-3D Generation with MVDream and SDXL	84
4.7	Novel View Synthesis with a Different Number of Reference and Target Views	88
5.1	Autoregressive Novel View Generation	92
5.2	Causal Multi-view Diffusion Pipeline	98
5.3	CaPE Attention Score Analysis	100
5.4	Novel View Synthesis Results	101
5.5	Novel View Synthesis on Diverse Customized Trajectories	103
5.6	Examples of Key Properties of CausNVS	108
5.7	Novel View Synthesis on DL3DV with Increasing Input Views -Part 1	110
5.8	Novel View Synthesis on DL3DV with Increasing Input Views -Part 2	110
5.9	Novel View Synthesis on Re10K Compared with Baselines - Part 1	111
5.10	Novel View Synthesis on Re10K Compared with Baselines - Part 2	111

Chapter 1

Introduction

Contents

1.1 Spatial AI	1
1.2 3D Representations	4
1.3 Generative Models	6
1.4 Towards World Modelling	8
1.5 Contributions	9
1.6 Publications	10
1.7 Thesis Structure	11

1.1 Spatial AI

Imagine a robot tasked with cleaning a cluttered room, fetching an object from a table, or navigating through a busy warehouse. To perform these tasks effectively, it must perceive the geometry of its surroundings, understand what objects are present, and predict how the world might change as it moves or acts. Some parts of the scene—such as the shape of a chair or a wall—can be inferred reliably from limited observations, while others, like complicated or partially occluded objects, require repeated inspection and refinement. An intelligent system capable of operating autonomously must

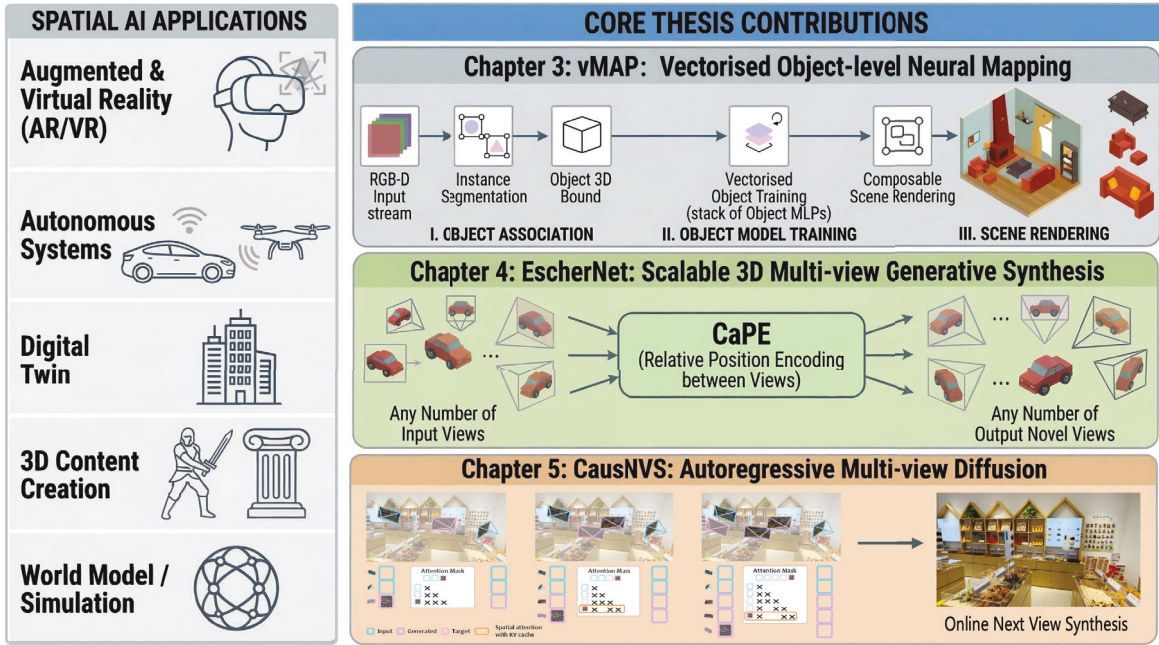


Figure 1.1: **Applications of Spatial AI and overview of the thesis contributions.** Spatial AI enables intelligent systems that perceive, reconstruct, and generate 3D environments, supporting applications such as augmented and virtual reality (AR/VR), autonomous systems, digital twins, 3D content creation, and world modelling and simulation. This thesis develops generative and neural mapping methods for scalable 3D scene understanding and synthesis. Chapter 3 introduces *vMAP*, a vectorised object-level neural mapping framework that decomposes scenes into composable object representations. Chapter 4 presents *EscherNet*, a scalable multi-view generative model that synthesises novel views from arbitrary input views using camera-aware positional encoding (CaPE). Chapter 5 proposes *CausNVS*, an autoregressive multi-view diffusion model that generates spatially and temporally consistent views via causal attention and key-value caching.

therefore build an internal model of the world that integrates prior knowledge with ongoing observations, updating its understanding as new evidence arrives. This capability, which we might call *spatial intelligence*, underpins autonomous navigation, robotic manipulation, augmented and virtual reality (AR/VR), digital twins, and large-scale world simulation.

Spatial Artificial Intelligence (Spatial AI) aspires to endow intelligent systems with this spatial understanding, the ability to perceive, map, and interact with the physical world in a structured, predictive, and semantically meaningful manner [Dav18]. Unlike traditional computer vision, which focuses on recognising isolated objects or classifying images, Spatial AI aims to construct persistent models of the environment that can support localisation, reconstruction, reasoning, and interaction. Such world models must not only represent what is currently seen but also allow the system to anticipate unseen parts of the environment and reason about the consequences of potential actions.

At the heart of Spatial AI lies the problem of representation: *how should the world be modelled to support perception, prediction, and interaction at scale?* Dense 3D reconstructions can capture fine geometric detail, but they are computationally expensive and redundant when applied uniformly to every region of the scene. Conversely, purely prior-driven models can hallucinate plausible but inaccurate geometry when observations are sparse or ambiguous. A truly capable Spatial AI system must therefore *adaptively allocate modelling capacity*, refining uncertain regions through additional observations while relying on priors for well-understood structures. In other words, it must learn when to infer, when to observe, and how to fuse information from both.

Such a system must also reason beyond static geometry. Actions taken by an agent, whether grasping an object, opening a door, or turning a corner, modify the environment and change its observable state. Hence, Spatial AI extends naturally toward *world modelling*: learning representations that are spatially grounded yet predictive of future outcomes.

This thesis explores how structured and generative representations can together provide the foundation for such models: representations that are geometrically precise, semantically meaningful, scalable to large environments, and capable of synthesising new, consistent views of the world. To this end, this dissertation develops three interconnected systems that push the boundaries of object-level mapping, view synthesis, and causal reasoning in 3D:

- **vMAP** (*Vectorised Object-level Neural Mapping*) tackles the “structured” challenge by introducing an object-centric SLAM system that represents scenes as a dynamic collection of vectorised, independent neural implicit fields, enabling efficient, editable, and semantic-aware mapping.
- **EscherNet** (*Scalable 3D Multi-view Generative Synthesis*) addresses the “generative and scalable” challenge, proposing a novel multi-view diffusion model that can synthesise any number of geometrically consistent novel views from any known views, enabling rich, prior-driven scene completion.
- **CausNVS** (*Autoregressive Multi-view Diffusion*) unifies mapping and generation into “causal world modelling” by introducing an autoregressive multi-view diffusion system that uses causal

memory to generate online sequences of novel views that are consistent in space and time.

Together, these contributions demonstrate that coupling structured 3D representations with generative modeling provides a scalable and robust foundation for building predictive world models for Spatial AI.

1.2 3D Representations

A central question in Spatial AI is: *how should the physical world be represented computationally?* The choice of representation governs how effectively a system can model its surroundings, how efficiently it can integrate new observations, and how well it can support reasoning, planning, and synthesis. Progress in Spatial AI has therefore been tightly coupled with the evolution of 3D representations.

Classical Representations. Most SLAM systems relied on explicit geometric structures such as point clouds, voxel grids, and polygonal meshes. Point clouds are efficient and directly obtained from depth sensors, but lack topological information and semantic structure. Voxel grids encode volumetric occupancy but scale cubically with scene size, making dense reconstructions computationally prohibitive for large or dynamic environments. Meshes explicitly represent surface geometry and remain central to graphics and reconstruction pipelines, yet they are difficult to update dynamically and less suited to differentiable learning. While these explicit forms underpin the success of classical SLAM [DRMS07, MAMT15, NIH⁺11], they are limited in expressiveness and struggle to integrate high-level semantics or predictive reasoning.

Neural Implicit Representations. Beyond classical discrete structures, implicit approaches encode 3D geometry as continuous functions parameterised by neural networks. Occupancy Networks [MON⁺19] learn a classifier over 3D space that predicts whether a point lies inside or outside an object, while DeepSDF [PFS⁺19] regresses a signed distance function whose zero-level set defines the surface. These neural implicit methods produce smooth, compact reconstructions unconstrained

by topology or discretisation, and are naturally optimised via gradient descent. However, they usually require ground-truth 3D supervision and focus solely on geometry, limiting their scalability.

Differentiable Rendering and Neural Fields. To reduce reliance on 3D supervision, differentiable rendering integrates a graphics pipeline into learning, allowing gradients to flow from images back to scene parameters. Neural Radiance Fields (NeRF) [MST⁺20a] represent a scene as a function mapping 3D coordinates and viewing directions to density and colour, synthesised through volumetric ray marching. While NeRF achieves photo-realistic novel view synthesis, its training and inference were initially prohibitively slow. Subsequent works improved efficiency by introducing explicit data structures such as voxel feature grids (Plenoxels [FKYT⁺22], DirectVoxGO [SSC22]), multi-resolution hash encodings (Instant-NGP [MESK22]), and point-based primitives (Point-NeRF [XXP⁺22]). More recently, 3D Gaussian Splatting (3DGS) [KKLD23] has emerged as a faster and more practical alternative, replacing per-ray volume integration with rasterisation and alpha-blending of semi-transparent Gaussian primitives. By exploiting scene sparsity and GPU-friendly operations, 3DGS delivers real-time, high-quality rendering and flexible scene updates. Together, these advances have expanded neural fields beyond appearance to surfaces and semantics, and they are now being actively explored as unified map representations for SLAM.

Object-level and Compositional Representations. To move beyond monolithic scene models, researchers have proposed object-centric and compositional representations. Such methods decompose scenes into sets of modular primitives, such as objects, surfaces, or Gaussians, each with learnable parameters and semantics. Object-centric SLAM systems such as SLAM++ [SMNS⁺13] and Quadric-SLAM [NMS18] demonstrated that representing the world as structured entities improves localisation and long-term mapping. More recent neural object-centric methods [SWD20, XLT⁺19] integrate learned descriptors and semantic embeddings, bridging geometric mapping with high-level reasoning. Such compositionality is essential for scalable and interpretable spatial understanding: it allows robots to reason about objects as functional entities, to reuse known structures across environments, and to update only the parts of the world that have changed.

Towards Generative Representations. Beyond reconstructing geometry, generative models have enabled the learning of 3D priors across diverse scenes. Instead of building every map from scratch, a system can infer plausible shapes and appearances from limited observations and refine them as more evidence becomes available. Models such as Zero-1-to-3 [LWVH⁺23] generate novel views from a single image by learning cross-scene priors through diffusion, bypassing explicit 3D reconstruction. This generative perspective marks a paradigm shift: from deterministic reconstruction toward probabilistic, data-driven world modelling that scales across scenes and tasks.

This thesis builds on these developments by pursuing both structured and generative representations. Chapter 3 introduces *vMAP*, a vectorised neural SLAM framework that models scenes as learnable object-level primitives, enabling scalable, parallel, and compositional mapping. Chapters 4 and 5 then explore diffusion and autoregressive generative models for scalable, world-consistent multi-view synthesis, laying the foundation for learned 3D priors that generalise across environments.

1.3 Generative Models

Generative modelling has fundamentally transformed visual understanding. Instead of merely reconstructing what is observed, generative models aim to capture the underlying distribution of possible scenes, enabling the synthesis of new, plausible, and coherent samples beyond the current observation. For Spatial AI, this shift is profound. It moves perception from a purely reactive process to one that is also *predictive and imaginative*: an intelligent system should not only recover what it sees but also infer what it has not yet seen, anticipate how a scene might appear from new viewpoints, and reason about its structure in unseen configurations. This ability to *generate* as well as to *understand* makes generative models a natural extension of Spatial AI.

From GANs to Diffusion. Early generative methods, such as Generative Adversarial Networks (GANs), demonstrated that neural networks could produce strikingly realistic images through adversarial training. However, GANs were inherently unstable, lacked an explicit probabilistic interpretation, and offered little control over the generation process. Diffusion models [SDWVG15, HJA20,

[SSDK⁺20](#), [RBL⁺22](#)] introduced a conceptually different and statistically grounded approach. By learning to reverse a progressive noise corruption process, diffusion models achieve stable likelihood-based training, high visual fidelity, and fine-grained conditioning flexibility. Their probabilistic nature allows learning expressive data priors that capture the variability and uncertainty inherent in visual scenes. Latent Diffusion Models (LDMs) [\[RBL⁺22\]](#) further improved efficiency by performing denoising in compact latent spaces, preserving semantic and structural information while greatly reducing computational cost. These characteristics, including stability, expressivity, controllability, and scalability, make diffusion a foundational tool for extending generative modelling from 2D imagery to 3D and spatial reasoning.

Generative 3D Reconstruction. Extending generative modelling into three dimensions unlocks new possibilities for Spatial AI. Traditional 3D reconstruction relies on dense, view-dependent observations to recover scene geometry. In contrast, generative models can infer plausible shapes and appearances from sparse inputs by leveraging learned cross-scene priors. This shift reframes mapping as a process of *inference under uncertainty* rather than direct geometric optimisation. Models such as Zero-1-to-3 [\[LWVH⁺23\]](#) demonstrate that a single image can suffice to generate novel views using learned diffusion priors, while multi-view diffusion and autoregressive frameworks extend this idea to collections of viewpoints, capturing spatial dependencies across perspectives. For an embodied agent, such generative 3D reasoning is essential: it allows the system to hypothesise unobserved geometry, fill in occlusions, and anticipate scene continuity, all of which are core elements of spatial understanding that go beyond passive reconstruction. Generative 3D models thus bridge perception and imagination, offering a scalable foundation for Spatial AI systems capable of operating across diverse and dynamic environments.

Challenges and Complementary Paradigms. Despite their expressiveness, diffusion-based 3D models face several key limitations. Their iterative denoising process, though powerful, is computationally intensive and often applied jointly across all target views or frames to maintain appearance consistency. This joint denoising strategy breaks the causal dependencies required for temporally or spatially sequential prediction, which are essential for agents that must reason about how the world

evolves. To overcome this limitation, *autoregressive* models provide a complementary approach. By generating frames or views sequentially, they naturally encode causal structure and enable temporally coherent rollouts, making them particularly suitable for predictive and interactive settings. However, purely autoregressive generation tends to accumulate drift over long sequences and struggles to match the visual fidelity achieved by diffusion. These two paradigms are therefore best viewed as complementary rather than competing: diffusion offers strong priors and high-quality synthesis, while autoregression brings causal reasoning and efficient sequential prediction. This thesis explores how the integration of both, combining diffusion’s generative richness with autoregression’s structural consistency, can yield scalable, coherent, and flexible multi-view generation.

1.4 Towards World Modelling

The convergence of generative and causal modelling naturally leads to the concept of a *world model*. A world model captures not only the static structure of an environment but also its dynamics and evolution over time. It allows an agent to simulate how its surroundings might change, to anticipate the outcomes of its actions, and to plan accordingly. In this sense, Spatial AI and generative modelling are deeply aligned, as both aim to build compact, generalisable, and temporally consistent representations of reality that support prediction, reasoning, and interaction.

Early World Models. The idea of learning internal models of the world was first explored in reinforcement learning, where agents developed compact latent dynamics models to imagine and plan future trajectories. Notable examples include the World Models framework [HS18] and the Dreamer series [HLBN20, HLN21, HPBL23], which train generative state-space models capable of predicting future states in latent space. These systems demonstrate that learned dynamics can replace explicit simulators, linking perception with prediction and enabling planning through imagination.

Large-scale Visual World Models. Recent advances have extended this principle to visual domains. Models such as Genie [BDE⁺24, PHBB⁺24] and GameNGen [VLA25] train autoregressive

Transformers to predict future frames from discrete visual tokens, achieving controllable, temporally coherent scene evolution. The latest generation, Genie 3 [BBB⁺25], scales this idea to real-time, interactive video-based environments, supporting agent actions and long-horizon consistency. These developments mark a significant step from short-term video prediction to embodied visual simulation, representing progress toward world models that unify perception, generation, and interaction.

Challenges. Building scalable and spatially grounded world models remains an open challenge. Maintaining geometric and semantic coherence across long temporal horizons requires representations that are both structured and generative. Further obstacles include efficient long-term memory, multimodal alignment, and the principled evaluation of predictive quality. This thesis connects to this emerging research direction by introducing CausNVS (Chapter 5), an autoregressive diffusion model for multi-view synthesis. By enforcing causal dependencies between reference and target views and maintaining spatially consistent rollouts, CausNVS embodies the philosophy of world modelling within the visual domain, using generative models not merely to reconstruct what is observed, but to imagine and predict what lies beyond.

1.5 Contributions

This thesis advances Spatial AI by unifying structured 3D representations and generative modelling. It aims to bridge the gap between geometric reconstruction and scalable, world-consistent synthesis. The key contributions are as follows:

- **Vectorised Object Mapping (vMAP).** We introduce *vMAP*, a neural SLAM framework that represents environments as sets of learnable vectorised primitives. By combining object-level decomposition with efficient vectorised training, vMAP achieves scalability, compositionality, and parallel optimisation, bridging the gap between dense neural SLAM and structured object-centric mapping.
- **Scalable Multi-view Diffusion (EscherNet).** We propose *EscherNet*, a diffusion-based generative model for view synthesis that learns cross-scene priors. Unlike scene-specific neural

rendering methods, EscherNet decouples synthesis from explicit coordinate systems by introducing a general relative camera pose encoding (CaPE) formulation to attention, enabling generalisation to unseen scenes, flexible conditioning on arbitrary numbers of views, and improved synthesis quality as more references are provided.

- **Autoregressive Multi-view Diffusion (CausNVS).** We present *CausNVS*, an autoregressive multi-view diffusion model that enables streaming view synthesis with arbitrary input-output configurations. Unlike prior approaches that denoise all target views simultaneously, CausNVS generates views sequentially with causal masking and per-frame noise, guided by pairwise-relative camera pose encodings (CaPE) for precise spatial control. At inference, a spatially-aware sliding window with key-value caching and noise conditioning augmentation maintains consistent rollouts efficiently. By combining autoregression with diffusion, CausNVS takes a step towards world models that require interactive streaming prediction with 3D consistency and efficient spatial memory.

1.6 Publications

The work described in this thesis resulted in the following publications:

- **Kong, X.,** Liu, S., Taher, M., Davison, A. J. (2023). *vMAP: Vectorised Object Mapping for Neural Field SLAM*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). [KLTD23]
Project Page: <https://kxhit.github.io/vMAP>
- **Kong, X.,** Liu, S., Lyu, X., Taher, M., Qi, X., Davison, A. J. (2024). *Eschernet: A Generative Model for Scalable View Synthesis*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) [Oral]. [KLL⁺24]
Project Page: <https://kxhit.github.io/EscherNet>
- **Kong, X.,** Watson, D., Strümler, Y., Niemeyer, M., Tombari, F. (2025). *CausNVS: Autoregressive Multi-view Diffusion for Flexible 3D Novel View Synthesis*. ArXiv. [KWS⁺25]

Project Page: <https://kxhit.github.io/CausNVS.html>

1.7 Thesis Structure

The remainder of this thesis is organised as follows:

- **Chapter 2** introduces the notions, camera models, 3D neural representations, SLAM, Transformers, diffusion, and autoregressive models. These concepts provide the background for the methods developed in later chapters.
- **Chapter 3** presents vMAP, a vectorised neural SLAM framework for scalable object-level mapping, introducing parallelised optimisation of neural primitives.
- **Chapter 4** presents EscherNet, a diffusion-based generative model that learns cross-scene priors for scalable and generalisable view synthesis with relative Camera Pose Encoding (CaPE).
- **Chapter 5** presents CausNVS, which combines autoregressive conditioning with diffusion to achieve causal, consistent, and flexible multi-view generation.
- **Chapter 6** summarises the contributions of this thesis, reflects on limitations, and outlines future directions in neural representations, generative modelling, and world-consistent Spatial AI.

Chapter 2

Preliminaries

Contents

2.1	Notation	13
2.2	Camera Model	14
2.3	Advanced 3D Representations	15
2.4	Visual SLAM	17
2.5	Transformers and Attention	21
2.6	Diffusion Models	23
2.6.1	Discrete Formulation (DDPM)	23
2.6.2	Continuous Formulation (SDE)	25
2.6.3	Conditioning and Guidance in Practice	25
2.6.4	Recent Formulations	26
2.6.5	Key Advanced Applications	27
2.7	Autoregressive Models	29
2.7.1	Attention Mask	29
2.7.2	Key Advanced Applications	31
2.7.3	Comparing Autoregressive and Diffusion Models	33
2.7.4	Towards World Models	34

2.1 Notation

We summarise the general notations in this thesis, that provide a common reference for 3D geometry, camera models, 3D neural representations, and generative models. In later chapters, additional symbols may be introduced or specialised according to the context of each proposed method.

- **3D point:** $\mathbf{x} \in \mathbb{R}^3$, a point in Euclidean space.
- **2D pixel:** $\mathbf{u} \in \mathbb{R}^2$, a pixel coordinate in the image plane.
- **Scene:** $\mathcal{S}$, a collection of 3D points, surfaces, or implicit functions representing geometry and appearance.
- **Image:** $\mathbf{I}$, an RGB image defined on a pixel grid $\mathbf{u} \in \mathbb{R}^2$.
- **Camera parameters:** $\mathcal{P} = (\mathbf{K}, \mathbf{T})$, consisting of intrinsic matrix $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ and extrinsics $\mathbf{T} \in SE(3)$, with rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$, mapping world to camera coordinates.
- **Neural representation:** f_θ , a neural field parameterised by θ (e.g., NeRF, DeepSDF, neural network).
- **Latent variable:** $\mathbf{z}$, a feature vector in learned representations.
- **Probability distribution:** $p(\mathbf{x})$ or $p_\theta(\mathbf{x}|\mathbf{z})$, with θ denoting learnable parameters.

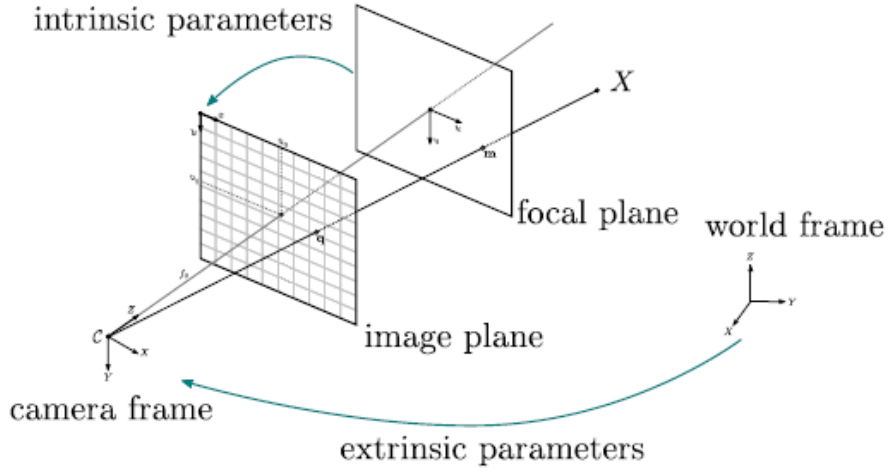


Figure 2.1: Illustration of the Pinhole Camera Model

2.2 Camera Model

We adopt the standard pinhole camera model. A 3D point $\mathbf{X} = (X, Y, Z)^\top \in \mathbb{R}^3$ in world coordinates is projected to a 2D pixel coordinate $\mathbf{u} = (u, v)^\top \in \mathbb{R}^2$ in the image plane by:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} [\mathbf{R} | \mathbf{t}] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (2.1)$$

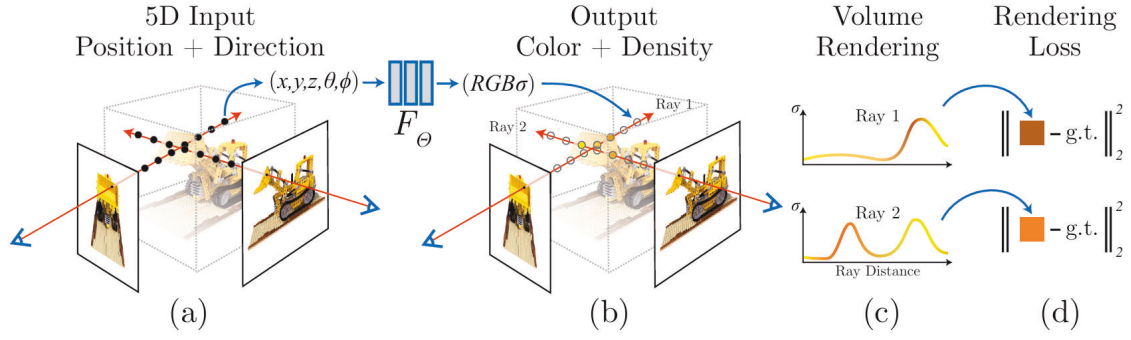
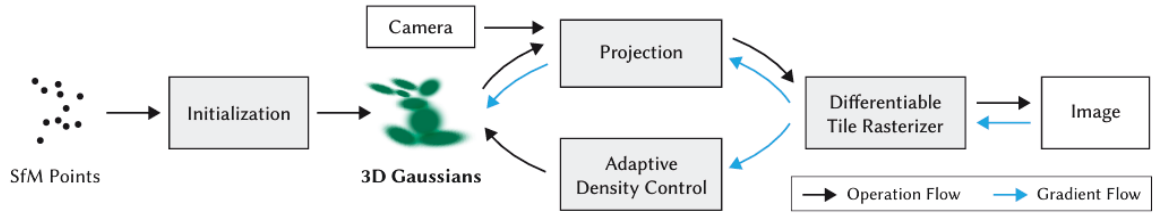
where $s \in \mathbb{R}$ is a scale factor, $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ is the intrinsic calibration matrix, and $[\mathbf{R} | \mathbf{t}]$ represents the extrinsic parameters with rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$.

The camera intrinsics are defined as:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.2)$$

where f_x, f_y are focal lengths and (c_x, c_y) is the principal point.

This model allows us to map between 3D geometry and image observations. It serves as the basis for

Figure 2.2: **Illustration of Neural Radiance Fields (NeRF)** [MST+20a]Figure 2.3: **Illustration of 3D Gaussian Splatting (3DGS)** [KKLD23]

view synthesis, 3D reconstruction, and camera-conditioned generative models, which are central to the methods proposed in this thesis.

2.3 Advanced 3D Representations

Representing 3D scenes in a compact and learnable form is a central challenge in computer vision and graphics. Classical scene representations such as meshes, point clouds, and voxel grids provide explicit geometry but often struggle to capture fine details or view-dependent appearance in a differentiable and optimisable manner. Recent advances have introduced both neural and hybrid representations that model geometry and appearance continuously, enabling learning-based 3D reasoning. This section reviews several key advanced 3D representations that underpin the methods developed in this thesis.

- **Deep Signed Distance Functions (DeepSDF)** [PFS+19]: Traditional signed distance functions (SDFs) represent geometry explicitly on regular grids. DeepSDF extends this concept by learning a continuous implicit function $f_\theta(\mathbf{x}) \mapsto s$ parameterised by a neural network, predicting the

signed distance of a point $\mathbf{x}$ to the nearest surface. The zero-level set $\{\mathbf{x} \mid f_\theta(\mathbf{x}) = 0\}$ defines the surface. DeepSDF provides watertight and topologically consistent reconstructions and has become a cornerstone for shape modelling, scene reconstruction, and learning geometric priors.

- **Neural Radiance Fields (NeRF)** [MST⁺20a]: As shown in Fig 2.2, NeRF models a continuous radiance field using a neural network $f_\theta(\mathbf{x}, \mathbf{d}) \mapsto (\mathbf{c}, \sigma)$, which maps a 3D position $\mathbf{x}$ and viewing direction $\mathbf{d}$ to a color $\mathbf{c}$ and a volume density σ . Images are rendered via volume rendering:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t) \sigma(\mathbf{r}(t)) \mathbf{c}(\mathbf{r}(t), \mathbf{d}) dt,$$

where $T(t)$ denotes transmittance. NeRF introduced a major breakthrough for photorealistic novel view synthesis, though it typically requires per-scene optimisation and dense sampling, limiting scalability.

- **Gaussian Splatting** [KKLD23]: Gaussian Splatting represents a scene explicitly as a set of anisotropic 3D Gaussians $\{(\mu_i, \Sigma_i, \alpha_i, \mathbf{c}_i)\}_{i=1}^N$, where μ_i is the mean, Σ_i the covariance, α_i the opacity, and $\mathbf{c}_i$ the color or feature vector. As shown in Fig 2.3, these Gaussians are projected onto the image plane and blended through alpha compositing to produce rendered images. It combines differentiable rendering with explicit geometry and achieves real-time rendering performance, making it particularly effective for large-scale or interactive visualisation.
- **Primitive-based Scene Representations**: Primitive-based approaches decompose a scene into structured components such as semantic segments [MBD24], objects [SWD20, XLT⁺19], and constructed hierarchical graphs with nodes [RGA⁺20, AHG⁺19]. Each primitive encodes local geometry and appearance attributes, enabling modular scene composition and object-centric reasoning. Such representations promote scalability and reuse across different spatial or semantic entities. Chapter 3 of this thesis builds on this concept by introducing a vectorised primitive representation for object-level SLAM, allowing efficient aggregation of geometric and semantic information.
- **Learned Generative 3D Representations**: Generative approaches learn cross-scene 3D priors that allow novel view synthesis directly from multi-view images without explicit reconstruction. Formally, a model $g_\theta(\mathbf{I}_{1:N}, \mathcal{P}_{1:N}, \mathcal{P}_{1:M}^{\text{target}}) \mapsto \hat{\mathbf{I}}_{1:M}$ predicts target views $\hat{\mathbf{I}}_{1:M}$ from input

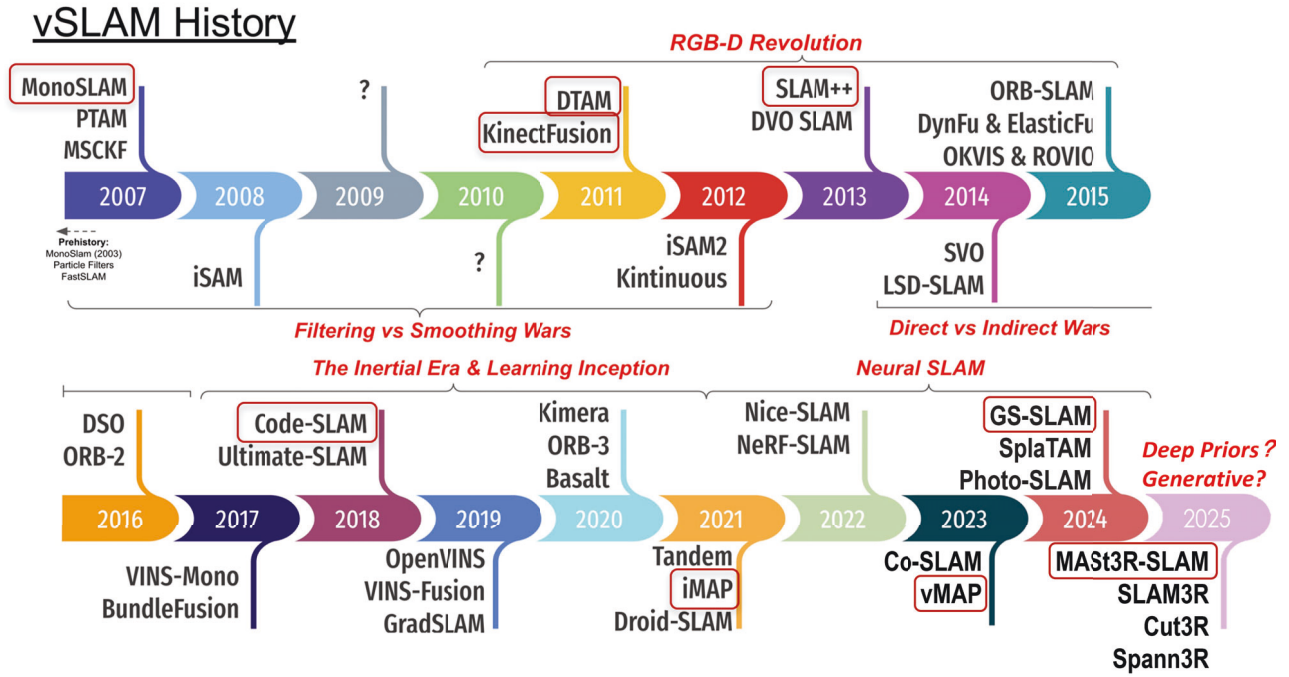


Figure 2.4: **Visual SLAM History** [Ros22]. Timeline of major visual SLAM systems, showing the transition from feature-based methods to dense mapping, semantic and object-level SLAM, neural implicit fields, and recent systems integrating learned priors.

images $I_{1:N}$ and their poses $\mathcal{P}_{1:N}$. These representations integrate implicit 3D reasoning within generative architectures, enabling generalisation across diverse scenes and training datasets. Chapter 4 presents *EscherNet*, which exemplifies this paradigm by learning a transferable 3D prior for scalable and generalisable multi-view synthesis. Chapter 5 further extends this direction with an autoregressive diffusion model for sequential view generation, where the attention KV cache functions as an implicit scene representation by accumulating rich spatial context across previously generated views.

2.4 Visual SLAM

Simultaneous localisation and mapping (SLAM) is a cornerstone of robotics and computer vision that continues to evolve over more than two decades. Its development reflects a continual pursuit of richer scene representations, scalability, and long-term consistency. Figure 2.4 illustrates this trajectory, from early sparse feature-based pipelines to recent semantic, object-level, and neural approaches.

Feature-based SLAM. Early visual SLAM systems primarily rely on sparse feature-based pipelines that combine geometric estimation and probabilistic filtering. The pioneering MonoSLAM [DRMS07] demonstrates the feasibility of real-time monocular SLAM by coupling feature tracking with an Extended Kalman Filter (EKF) for joint camera pose and map estimation. PTAM [KM09] further decouples tracking and mapping into parallel threads, improving robustness and scalability, particularly for augmented reality. Later, ORB-SLAM [MAMT15] and its successors ORB-SLAM2 [MAT17] and ORB-SLAM3 [CER⁺21] establish the modern standard for keypoint-based systems, integrating feature extraction, loop closure, and global optimisation in a unified framework. In parallel, direct methods such as LSD-SLAM [ESC14] and DSO [EKC17] replace discrete keypoints with pixel intensity alignment, achieving dense photometric tracking and better reconstruction in texture-rich regions. Together, these works define the algorithmic and system foundations of visual SLAM, balancing accuracy, robustness, and efficiency. However, their resulting maps are typically sparse or semi-dense, offering limited geometric or semantic detail for higher-level understanding.

Dense RGB-D SLAM. The advent of commodity RGB-D sensors, such as the Microsoft Kinect, sparks a transition toward dense geometric reconstruction. KinectFusion [NIH⁺11] introduces volumetric fusion based on truncated signed distance functions (TSDFs), with GPU acceleration enabling real-time dense mapping of small indoor scenes. DTAM [NLD11] demonstrates dense photometric tracking using variational optimisation on per-pixel depth maps, removing dependence on sparse keypoints. Subsequent systems push this further toward scalability and global consistency. ElasticFusion [WLSM⁺15] introduces a surfel-based representation with non-rigid loop closure, enabling consistent global reconstruction without explicit pose graphs, while BundleFusion [DNZ⁺17] combines real-time surface fusion with global pose graph optimisation to maintain map consistency over long trajectories. These dense RGB-D systems mark a major leap in visual SLAM—from sparse or semi-dense maps to truly dense, metrically accurate 3D reconstructions—but remain constrained by cubic memory growth and the computational cost of large-scale optimisation, while the limited use of priors often results in missing surfaces or holes in unobserved regions.

SLAM with Advanced 3D Representations. With the rise of learning-based 3D representations, SLAM begins to incorporate differentiable and continuous mapping functions. Neural implicit field methods such as iMAP [SLOD21], NICE-SLAM [ZPL⁺22], and Co-SLAM [WWA23] jointly optimise camera poses and neural scene representations online, producing continuous, photo-realistic reconstructions. More recently, 3DGS-SLAM [MMKD24, YQX⁺24, KKJ⁺24] leverages Gaussian primitives [KKLD23] for efficient, real-time dense mapping. These systems highlight the expressive power of differentiable 3D representations but remain limited by heavy per-scene optimisation and high memory demands, motivating subsequent work toward more compositional object-aware formulations and the integration of cross-scene priors.

Semantic and Object-level SLAM. To incorporate semantic structure and compositionality, SLAM systems evolve toward object-centric mapping. SemanticFusion [MHDL17] integrates CNN-based semantic segmentation with dense TSDF maps, while Co-Fusion [RA17a], MaskFusion [RBA18], and MID-Fusion [XLT⁺19] extend this idea to instance-level multi-object reconstructions with dynamic tracking. With richer semantics obtained by vision models, Feature Realistic Mapping [MSD23] fuses 2D image features into a 3D implicit map of high-dimensional embeddings, enabling semantic querying beyond one-hot class labels. Parallel efforts explore parametric object representations: SLAM++ [SMNS⁺13] pioneers the reuse of known CAD models for semantic mapping, while Quadric-SLAM [NMS18] and CubeSLAM [YS19] model objects as quadrics or cuboids to balance compactness and interpretability. Fusion++ [MCB⁺18] combines instance segmentation with volumetric fusion, and NodeSLAM [SWD20] integrates category-level priors by initialising each detected object with learned VAE latents and further optimising with the current observations. These advances push SLAM toward long-term, semantically meaningful, and compositional map representations. Within this trajectory, *vMAP* (Chapter 3) extends object-level SLAM by introducing vectorised neural primitives and efficient parallel training, bridging dense neural SLAM and object-centric mapping.

Integrating Large-scale Pre-trained Priors. A very recent trend is to augment SLAM with large-scale learned priors [WLC⁺24]. MAST3R-SLAM [MDD25] builds on the MAST3R [LCR24] two-view reconstruction prior, embedding learned pointmap predictions directly into a real-time monocular

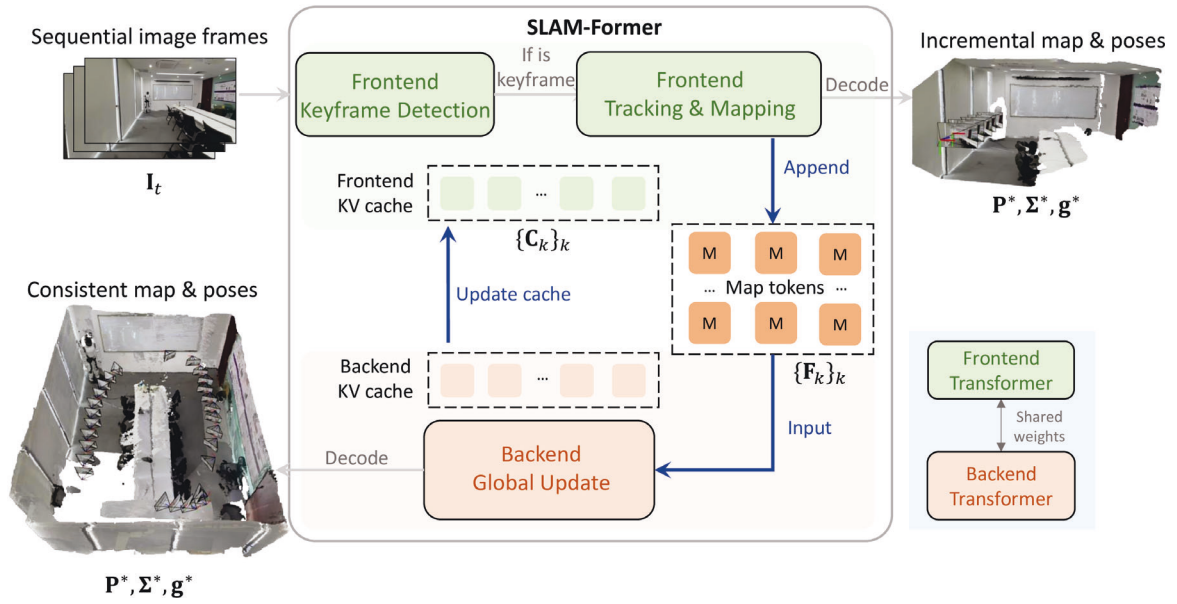


Figure 2.5: **Towards Transformer based SLAM** [YCL⁺25]. A SLAM-former type system unifies pose estimation and dense mapping within a single transformer-based model with key-value cache as spatial memory, and integrates pre-trained geometric priors seamlessly.

ular dense SLAM pipeline. By leveraging this geometric prior, MAST3R-SLAM achieves efficient tracking and dense mapping without fixed camera intrinsics, and supports robust loop closure and global optimisation. Similarly, systems such as SLAM3R [LDW⁺25] and VGGT-SLAM [MLC25] integrate pre-trained Dust3R [WLC⁺24] and VGGT [WCK⁺25] geometric priors to build dense point-based SLAM. In the meanwhile, Cut3R [WZH⁺25] and Spann3R [WA24] are recent transformer-based online 3D reconstruction models that leverage stateful or spatial memory to incrementally integrate information from image streams, enabling temporally consistent and dense scene reconstruction without explicit alignment. These approaches exemplify a paradigm shift: rather than relying purely on online optimisation and explicit feature matching, SLAM increasingly couples with powerful feed-forward priors. Looking forward, tighter integration with generative models and transformer architectures is anticipated. As illustrated in Fig. 2.5, a SLAM-former [YCL⁺25] paradigm, equipped with spatial attention and key-value memory, unifies mapping and localisation within a single feed-forward transformer.

Summary. The evolution of SLAM reflects a continuous expansion—from sparse geometric maps to dense volumetric reconstructions, to differentiable neural fields, to structured and object-level representations, and now toward systems enriched with large-scale priors and generative reasoning. Each

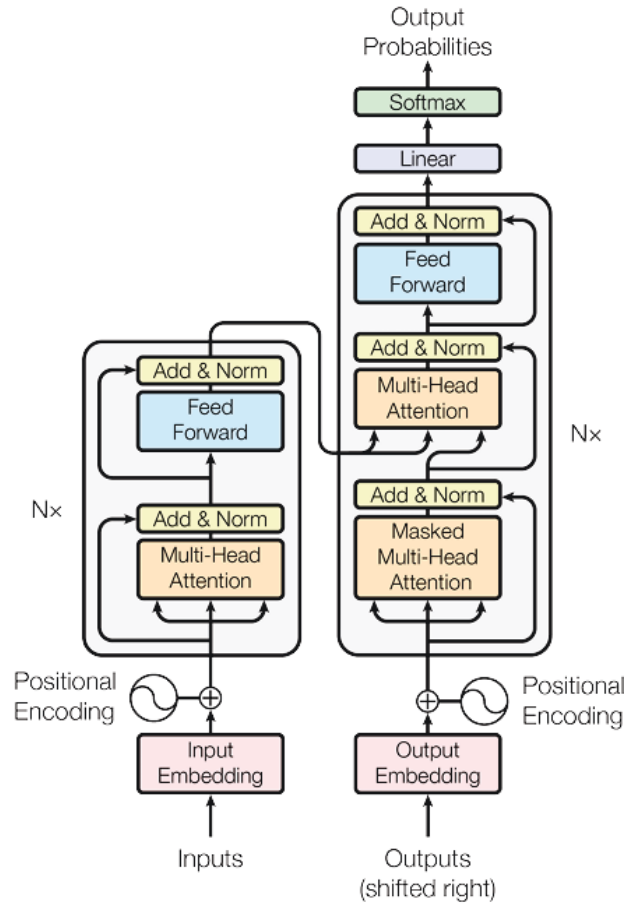


Figure 2.6: **The Transformer Architecture** [VSP⁺17]. The encoder (left) uses stacked multi-head attention and feed-forward layers. The decoder (right) includes masked self-attention for autoregressive generation and cross-attention to the encoder outputs.

stage is driven by the same underlying goals: richer and more expressive scene representations, improved scalability, and stronger long-horizon consistency. This thesis builds upon this trajectory: *vMAP* addresses scalability at the object level, while subsequent chapters explore diffusion- and autoregression-based generative models as steps toward world-consistent Spatial AI.

2.5 Transformers and Attention

Transformers [VSP⁺17] have become the dominant architecture for sequence modelling and generative tasks. Their power comes from the *attention mechanism*, which allows information to be aggregated from all elements in a sequence, capturing long-range dependencies more effectively than recurrent or convolutional models.

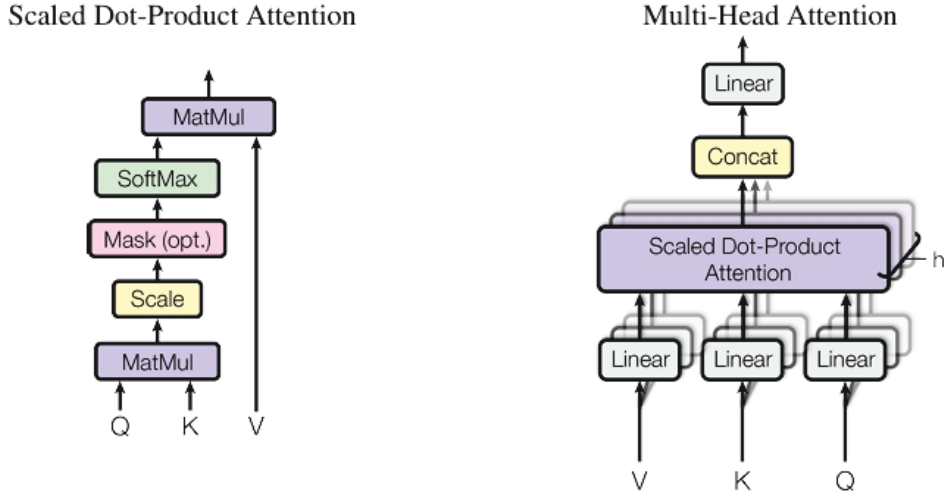


Figure 2.7: **Illustration of Attention** [VSP⁺17]. Scaled dot-product attention (left) and multi-head attention (right). Multiple heads project the input into different subspaces and jointly capture diverse relations.

Transformer Architecture Transformers [VSP⁺17] have become the dominant architecture for sequence modelling and generative tasks. Their power comes from the *attention mechanism*, which allows information to be aggregated from all elements in a sequence, capturing long-range dependencies more effectively than recurrent or convolutional models. Unlike fixed receptive fields in convolutional networks or sequential information propagation in recurrent models, attention enables each token to dynamically attend to the most relevant parts of the input. This adaptive global interaction allows Transformers to model complex dependencies while maintaining efficient parallel computation.

Attention Mechanism Given an input sequence $\mathbf{X} \in \mathbb{R}^{n \times d}$, attention computes pairwise interactions by projecting $\mathbf{X}$ into *queries*, *keys*, and *values*:

$$\mathbf{Q} = \mathbf{X}W_Q, \quad \mathbf{K} = \mathbf{X}W_K, \quad \mathbf{V} = \mathbf{X}W_V,$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d \times d_h}$ are learnable matrices. The scaled dot-product attention (Figure 2.7) is defined as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_h}}\right) \mathbf{V}.$$

Here, the similarity $\mathbf{Q}\mathbf{K}^\top$ measures relevance between elements, softmax normalises the weights, and the output is a weighted combination of values.

Multi-Head Attention Rather than relying on a single attention distribution, the Transformer employs multiple heads in parallel:

$$\text{MultiHead}(\mathbf{X}) = \text{Concat}(\text{head}_1, \dots, \text{head}_H)W_O,$$

where each $\text{head}_i = \text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i)$ uses its own projections. This enables the model to jointly attend to information at different positions and represent multiple types of dependencies.

Attention is a key building block in our methods. In Chapter 4, attention enables the interaction between reference and target views in a multi-view diffusion model. In Chapter 5, masked attention implements autoregressive conditioning, allowing flexible generation of novel views with causal dependencies across multiple inputs.

2.6 Diffusion Models

Diffusion models [SDWMG15, HJA20, SSDK⁺20] learn a data distribution by reversing a gradual noising process. A clean sample is progressively perturbed toward a simple noise prior, and a neural network is trained to invert this process step by step. We first introduce the discrete-time denoising diffusion probabilistic model (DDPM) [HJA20] and continuous-time stochastic differential equation (SDE) [SSDK⁺20] formulations, then summarise practical training and conditioning choices, and finally highlight key advances most relevant to this thesis.

2.6.1 Discrete Formulation (DDPM)

Forward (diffusion) process. Starting from $\mathbf{x}_0 \sim p_{\text{data}}$, define a T -step Markov chain:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}), \quad t = 1, \dots, T,$$

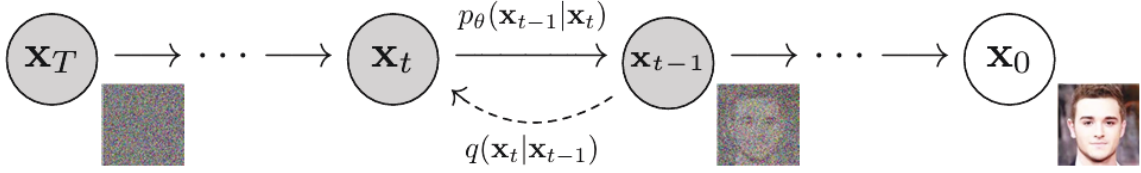


Figure 2.8: **Diffusion DDPM Formulation.** The forward chain $q(\mathbf{x}_t | \mathbf{x}_{t-1})$ gradually adds noise, and the reverse chain $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$ learns to denoise step by step.

with a variance schedule $\{\beta_t\}_{t=1}^T$. This yields the closed form:

$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}), \quad \alpha_t = 1 - \beta_t, \quad \bar{\alpha}_t = \prod_{s=1}^t \alpha_s.$$

As $t \rightarrow T$, $\mathbf{x}_T$ approaches a Gaussian prior.

Reverse (denoising) process and training. A neural network predicts the noise at each step, $\epsilon_\theta(\mathbf{x}_t, t)$, from which the reverse transition is parameterised as

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)).$$

The common training objective is the simplified denoising score matching loss:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{\mathbf{x}_0, t, \epsilon \sim \mathcal{N}(0, \mathbf{I})} \|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t)\|_2^2.$$

Representative parameterisations predict the noise (ϵ -prediction), the clean signal ($\mathbf{x}_0$ -prediction), or a velocity-like target (v -prediction). Different choices trade off stability, sharpness, and guidance behaviour.

Sampling variants. Beyond DDPM sampling, non-Markovian samplers such as DDIM [SME21] perform deterministic or semi-deterministic updates that reduce the number of steps with limited degradation. Figure 2.8 summarises the discrete forward and reverse processes.

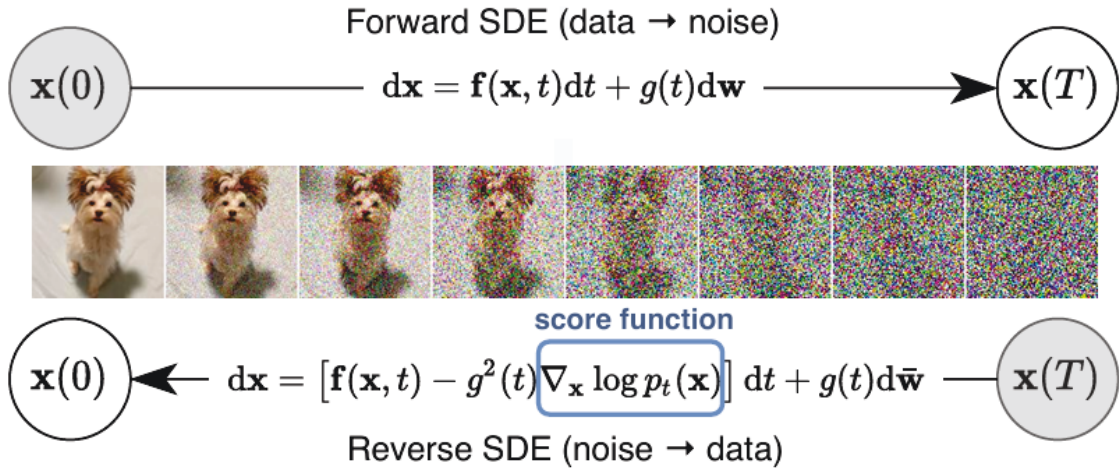


Figure 2.9: **Diffusion SDE Formulation.** The forward SDE maps data to noise, and the reverse SDE reconstructs samples guided by the learned score function.

2.6.2 Continuous Formulation (SDE)

Forward and reverse SDEs. In continuous time, diffusion can be written as a stochastic differential equation (SDE):

$$d\mathbf{x} = f(\mathbf{x}, t) dt + g(t) d\mathbf{w},$$

which transports data to a tractable noise distribution over $t \in [0, T]$. Sampling is performed by integrating the reverse-time SDE:

$$d\mathbf{x} = [f(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})] dt + g(t) d\bar{\mathbf{w}},$$

where $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ is the time-dependent score. In practice this score is learned by a neural network trained with score matching. Popular families include VE/VP SDEs [KAAL22] and probability-flow ODEs [SSDK⁺20] that share the same marginals as the reverse SDE but allow deterministic integration.

2.6.3 Conditioning and Guidance in Practice

Conditional diffusion. For conditional generation, one models $p_{\theta}(\mathbf{x}_0 \mid \mathbf{c})$ where the condition $\mathbf{c}$ can be text, class labels, images, or camera poses. In U-Net backbones [RFB15], conditioning is injected via cross-attention, FiLM-style feature modulation [PSDV⁺18], or time-dependent embeddings that

interact with intermediate feature maps.

Classifier-free guidance. A widely used technique interpolates between conditional and unconditional predictions [HS22] is:

$$\hat{\epsilon} = \epsilon_{\theta}(\mathbf{x}_t, \mathbf{c}, t) + s(\epsilon_{\theta}(\mathbf{x}_t, \mathbf{c}, t) - \epsilon_{\theta}(\mathbf{x}_t, \emptyset, t)),$$

with scale $s \geq 0$ controlling adherence to the condition versus sample diversity. This simple trick improves faithfulness with minimal architectural changes.

2.6.4 Recent Formulations

Recent works reinterpret diffusion through the lens of probability flows and consistency, aiming to improve training or sampling efficiency while preserving sample quality:

- **Flow Matching (FM)** [LCBH⁺23]: learns a vector field that transports a simple prior to data via a continuous-time ODE. Unlike SDE-based diffusion models, FM directly fits the deterministic probability flow while allowing flexible probability paths between data and noise during training.
- **Rectified Flow (RF)** [LGL23]: proposes simplified flows and training objectives that often accelerate convergence and reduce sampling steps.
- **Consistency Models (CM)** [SDCS23]: enforce step-to-step consistency to permit one- or few-step generation, trading some optimality for speed.

These formulations broaden diffusion to a family of flow-based generative methods. In practice, most large-scale vision and 3D systems relevant to this thesis still adopt latent diffusion backbones with the practical conditioning recipe above.

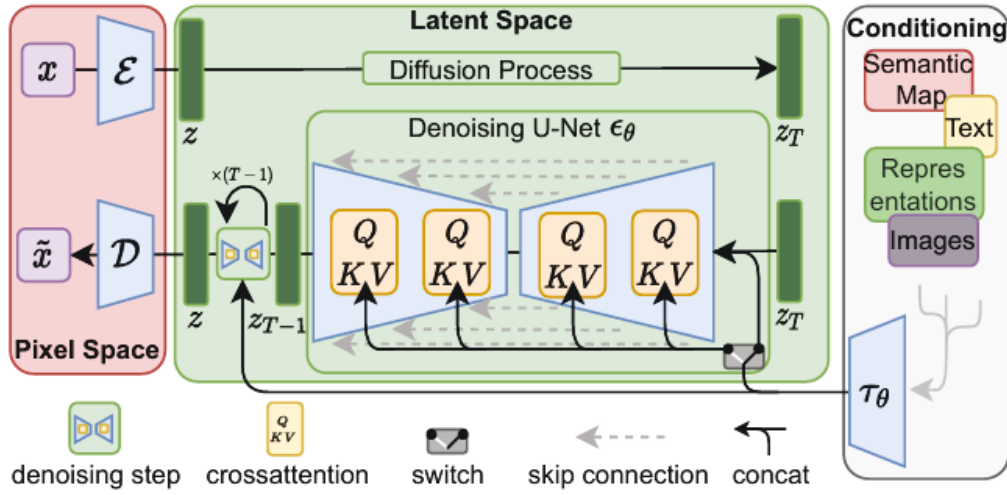


Figure 2.10: **Latent Diffusion Model** [RBL⁺22]. Instead of operating in pixel space, image diffusion is performed in a compressed latent space, with conditioning (e.g., text, semantic maps, images) injected via cross-attention, which substantially reduces computational requirements.

2.6.5 Key Advanced Applications

Latent Diffusion Models. Operating diffusion directly in pixel space is costly. Latent Diffusion Models [RBL⁺22] compress images with an autoencoder [KW14] ($\mathcal{E}, \mathcal{D}$) and run diffusion over $\mathbf{z} = \mathcal{E}(\mathbf{x})$:

$$\mathbf{z}_T \rightarrow \mathbf{z}_{T-1} \rightarrow \cdots \rightarrow \mathbf{z}_0, \quad \tilde{\mathbf{x}} = \mathcal{D}(\mathbf{z}_0).$$

This drastically reduces memory and compute while preserving fidelity, and supports rich conditioning via cross-attention inside the denoising U-Net [RFB15]. Figure 2.10 depicts the overall framework.

Video Diffusion Models. Extending diffusion to video requires maintaining temporal coherence in addition to spatial quality. Early extensions such as Video Diffusion Models (VDM) [HSG⁺22] applied diffusion directly to 3D (spatio-temporal) U-Nets but often suffered from motion inconsistency and temporal flicker. Stable Video Diffusion (SVD) [BDK⁺23] improved scalability and smoothness by performing latent-space diffusion with temporal attention layers, effectively coupling motion across frames. Complementary to these architectural advances, Align Your Latents (AYL) [BRL⁺23] fine-tunes latent diffusion with temporal alignment, ensuring that noise trajectories remain consistent across frames. This transforms per-frame denoising into a structured spatio-temporal process

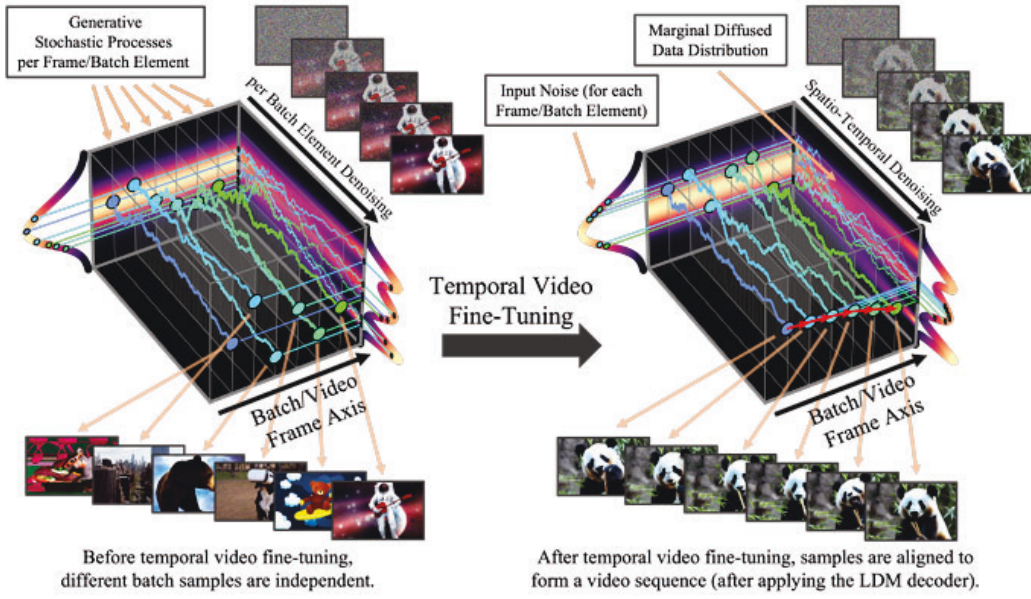


Figure 2.11: **Align Your Latents Video Diffusion Model** [BRL⁺23]. By jointly denoising multiple target frames, the model enforces consistency across frame-wise latent diffusion, thereby enabling temporally coherent video synthesis.

(Fig. 2.11), substantially reducing flicker and motion drift while preserving local details.

Novel View Synthesis Diffusion Models. Diffusion has been adapted to generate novel views conditioned on input images and target camera poses. As shown in Fig. 2.12, Zero-1-to-3 [LWVH⁺23] trains a latent diffusion model to sample:

$$p_{\theta}(\mathbf{I}_{\text{target}} \mid \mathbf{I}_{\text{ref}}, \mathcal{P}_{\text{ref}}, \mathcal{P}_{\text{target}}),$$

bypassing explicit 3D reconstruction by directly learning the conditional view distribution. Beyond single-view conditioning, several works extend diffusion to multi-view consistency. SyncDreamer [LLZ⁺24], MVDream [SWY⁺24] and CAT3D [GHH⁺24] jointly denoise multiple views to enforce geometric coherence, while MV-Diffusion [SWY⁺24] achieves globally consistent multi-view synthesis by sampling from a shared 3D latent representation.

Diffusion provides the generative backbone for the methods developed in this thesis. Chapter 4 adopts latent diffusion with multi-view conditioning to enable scalable view synthesis across scenes, and Chapter 5 combines diffusion with masked attention for autoregressive multi-view generation under flexible camera poses.

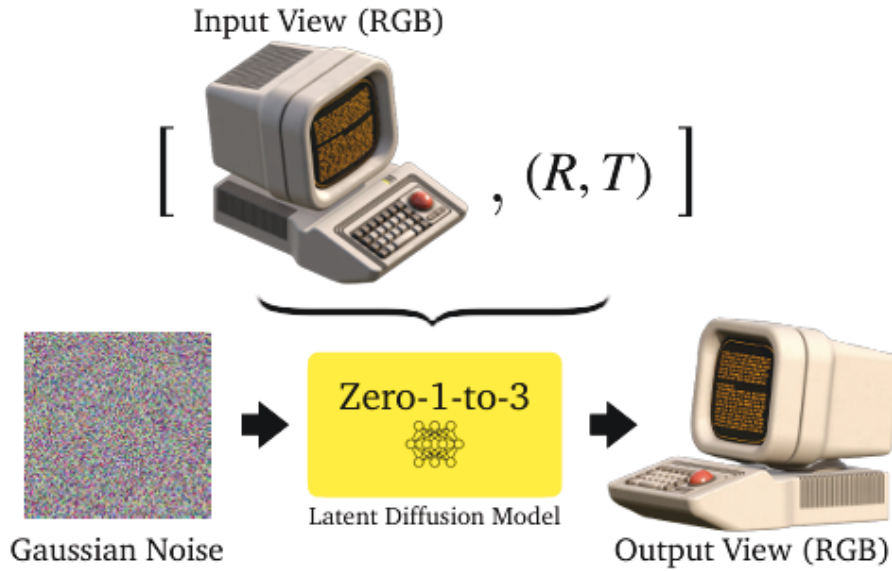


Figure 2.12: **Zero-1-to-3 View Synthesis Diffusion Model** [LWVH⁺23]. Novel view synthesis diffusion model conditioned on a single view and the relative pose.

2.7 Autoregressive Models

Autoregressive (AR) models constitute one of the most successful paradigms in generative modelling. They are particularly well suited for problems where outputs are generated sequentially with causal dependencies, which is relevant for the view-by-view generation setting studied later in this thesis. They decompose a joint distribution into a product of conditional probabilities, illustrated in Fig. 2.13, allowing exact likelihood evaluation and effective modelling of sequential dependencies:

$$p(x_1, x_2, \dots, x_n) = \prod_{i=1}^n p(x_i \mid x_{<i}),$$

where $x_{<i}$ denotes all previous elements. This factorisation underpins many breakthroughs across natural language, vision, video, and 3D domains.

2.7.1 Attention Mask

AR models are typically implemented with recurrent networks or Transformers. The key mechanism in modern architectures is masked self-attention, which ensures that each position can only attend to

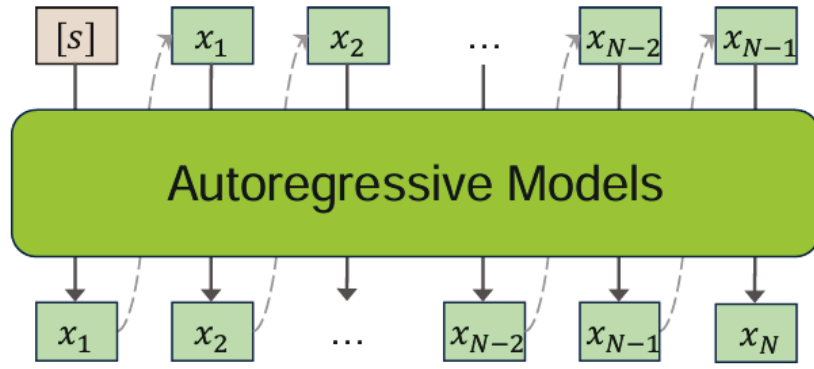


Figure 2.13: **Autoregressive Sequence Modeling** [XLH⁺25]. The joint distribution is factorised into conditionals, where each token x_i is generated conditioned on all its predecessors $x_{<i}$.

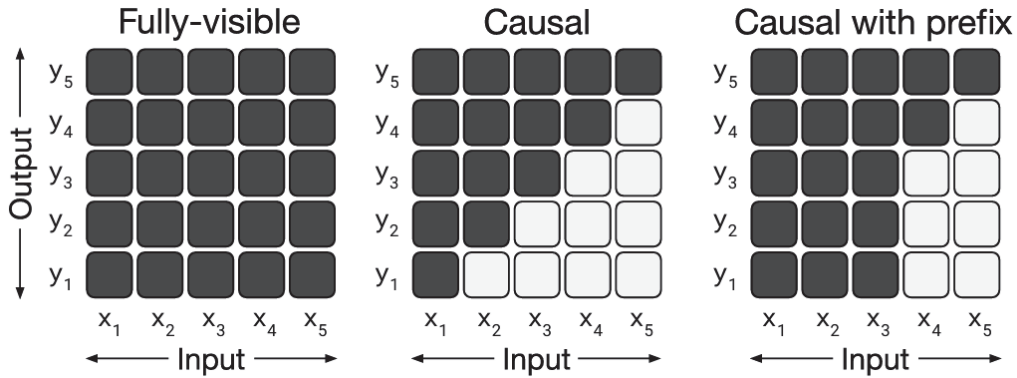


Figure 2.14: **Different Attention Mask Patterns** [RSR⁺20]. A schematic of full attention, causal masking, and prefix masking. A causal mask used in the transformer prevents access to future tokens, ensuring that each output is conditioned only on the past.

past tokens. The attention operation with a causal mask $\mathbf{M}$ is

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_h}} + \mathbf{M}\right) \mathbf{V},$$

where $\mathbf{M}$ assigns $-\infty$ to positions corresponding to future tokens. This masking enforces autoregression while preserving parallel computation during training. Figure 2.14 illustrates the causal masking mechanism.

Recent developments further extend AR attention with mechanisms that address the quadratic complexity of full attention. **Sparse attention** restricts each token to attend only to a subset of positions, such as local windows and periodic strides, as in the Sparse Transformer [CGRS19], Longformer [BPC20], and BigBird [ZGD⁺20]. **Linear attention** replaces the softmax kernel with feature maps that yield attention computation linear in sequence length, with examples including the Per-

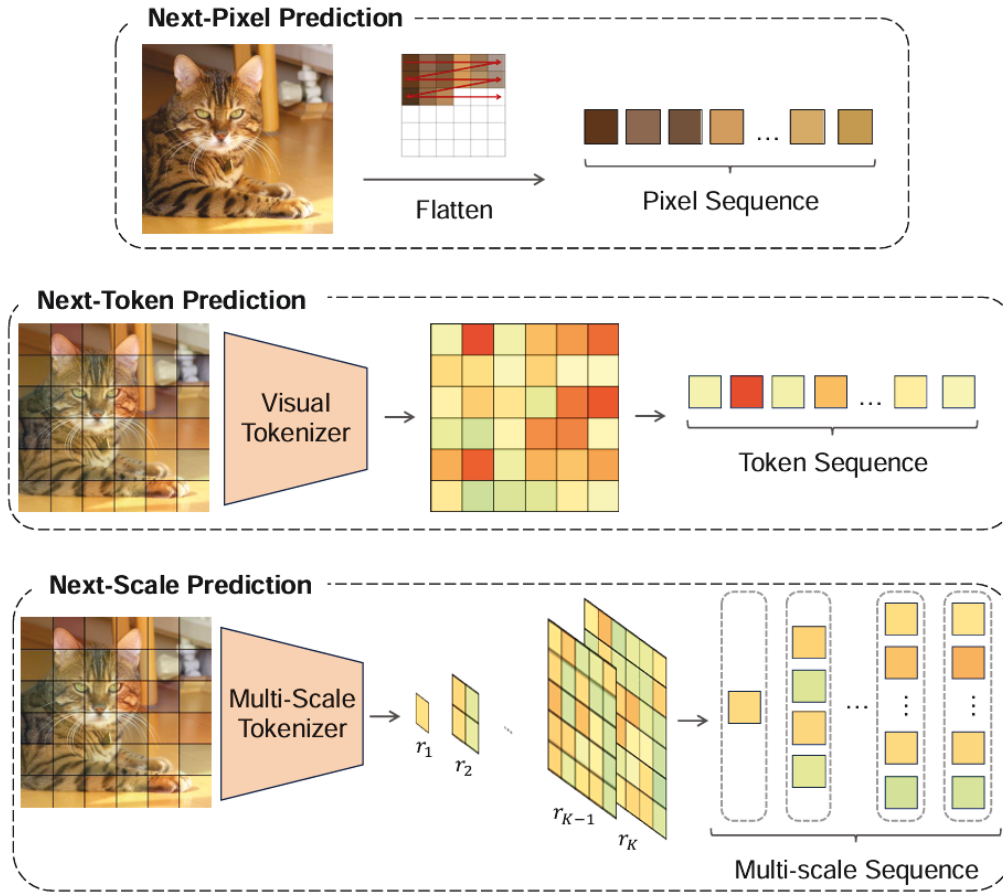


Figure 2.15: **Three Types of Visual Autoregressive Models** [XLH⁺25]. Next-Pixel Prediction flattens the image into a pixel sequence. Next-Token Prediction converts the image into a token sequence via a visual tokenizer. Next-Scale Prediction employs a multi-scale tokenizer to generate a multi-scale sequence.

former [CLD⁺21]. **Retrieval-augmented attention** augments the context with an external memory of relevant past tokens or documents, as in REALM [GLT⁺20] and Retro [BMH⁺22]. These innovations have enabled Transformers to scale from thousands to millions of tokens, and have been crucial for the emergence of large-scale autoregressive models in both language and multimodal domains.

2.7.2 Key Advanced Applications

Language The earliest successes of AR were in natural language processing. N-gram models already followed the AR principle [BDPD⁺92], but suffered from sparsity and limited context. Neural AR models based on RNNs [MKB⁺10] and LSTMs [HS97] alleviated some of these issues but were difficult to train and scale. The introduction of the Transformer [VSP⁺17] marked a breakthrough: its masked attention mechanism enabled scalable AR modelling across long sequences. The GPT se-

ries [RNS⁺18, BMR⁺20a] scaled this architecture to unprecedented levels, leading to large language models (LLMs) that exhibit emergent reasoning and generative abilities. This success established AR Transformers as the backbone of modern language generation.

Image The success of AR in text inspired analogous approaches in vision. Three types of visual autoregressive frameworks have been explored so far as illustrated in Fig 2.15. Early works such as PixelRNN [VDOKK16] and PixelCNN [VdOKE⁺16] adopted a *next-pixel prediction* paradigm, where an image is flattened into a sequence of pixels and each pixel is generated conditional on the previous ones. This allowed exact likelihood training and strong density estimation, but the sequential nature of pixel-level generation made sampling prohibitively slow for high-resolution images.

Subsequent research shifted to a *next-token prediction* paradigm. Instead of raw pixels, images are tokenised into discrete visual tokens using vector quantisation or learned tokenisers. ImageGPT demonstrated that GPT-style AR Transformers over such tokens could generate coherent and diverse images, showing that AR models could rival GANs and diffusion in image quality [CRC⁺20]. This formulation significantly reduced sequence length compared to pixels, improving training efficiency.

More recently, *next-scale prediction* has been explored, where multi-scale tokenisers break an image into hierarchies of patches or tokens, and the model autoregressively predicts across scales [TJY⁺24]. This design balances fidelity and efficiency: coarse tokens capture global structure, while finer tokens refine local details. Multi-scale AR methods have achieved state-of-the-art trade-offs between likelihood, quality, and speed.

Video Extending AR to video requires modelling both spatial quality and temporal coherence. Early works such as VideoGPT [YZAS21] extended autoregressive Transformers to spatio-temporal tokens, demonstrating coherent video generation from quantised VQ-VAE representations. Building on this, MAGVIT [YCS⁺23] and its successor MAGVIT-v2 [YLG⁺24] introduced masked token prediction for more efficient video AR. Other approaches adapt MaskGIT-style parallel decoding to video [CZJ⁺22], reducing sequential bottlenecks while retaining temporal consistency. Despite progress, challenges remain in scaling AR to long video horizons, where causal dependency mod-

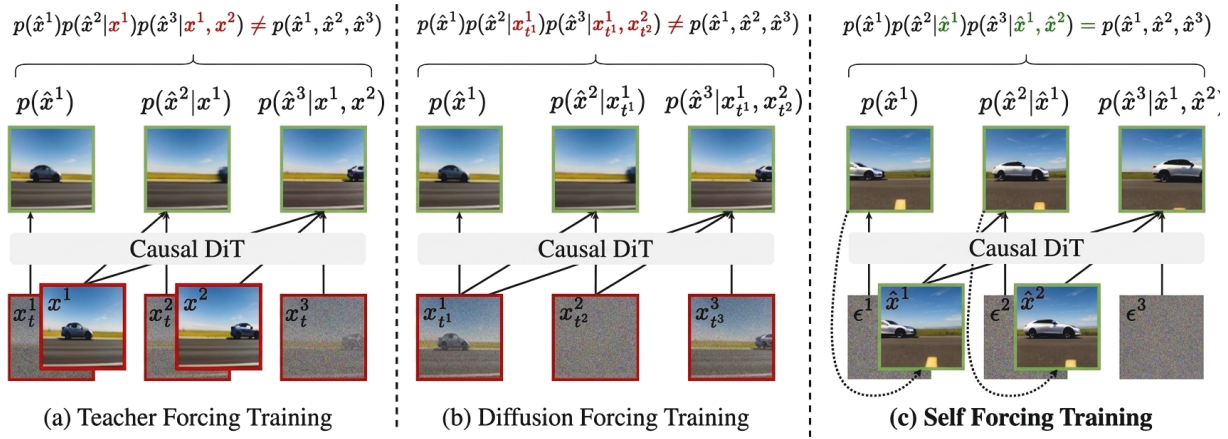


Figure 2.16: **Teacher Forcing, Diffusion Forcing, and Self-Forcing** [HLH⁺25]. Three ways of training autoregressive models: (a) Teacher forcing conditions on ground-truth past tokens, but suffers from exposure bias. (b) Diffusion forcing conditions on noisy versions of ground-truth tokens, blending AR with diffusion-style denoising. (c) Self-forcing conditions on the model’s own predictions, achieving consistency between training and inference.

elling is beneficial but computationally demanding.

2.7.3 Comparing Autoregressive and Diffusion Models

AR and diffusion are the two dominant paradigms in modern generative AI. They have complementary strengths and limitations:

- **Autoregression:** provides tractable likelihoods, straightforward training via maximum likelihood, and strong causal dependency modelling. However, sampling is inherently sequential and thus slow, especially for high-dimensional data.
- **Diffusion:** generates in parallel across dimensions per step, supports flexible conditioning, and achieves state-of-the-art fidelity. However, diffusion models lack tractable likelihoods and require many denoising steps at inference.

Hybrid approaches are emerging to combine the strengths of AR and diffusion. MaskGIT [CZJ⁺22] introduces parallel masked AR prediction, while other lines of work explicitly bridge the two paradigms. Diffusion Forcing [CMMD⁺24] reformulates AR decoding as iterative refinement, where predictions are gradually denoised in a diffusion-like manner. More recently, Self-Forcing [HLH⁺25] augments

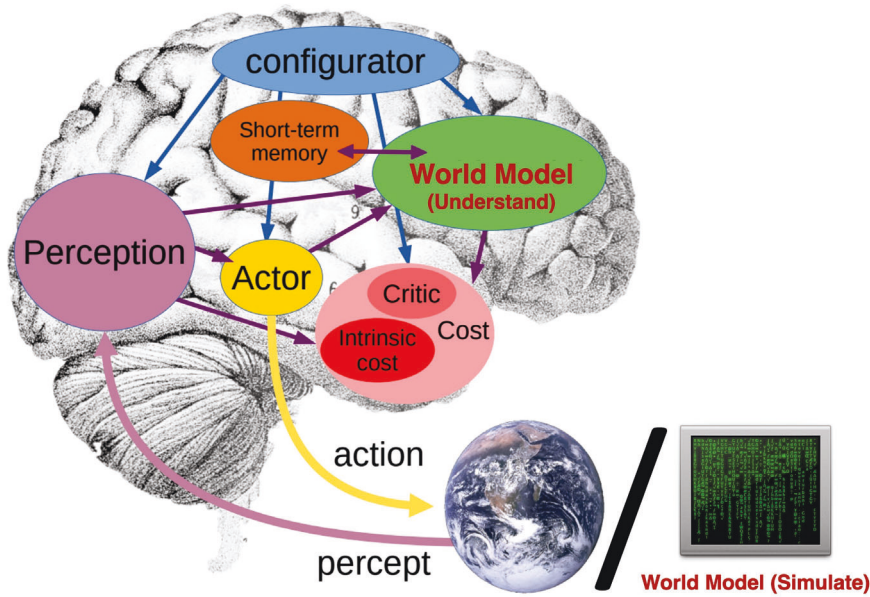


Figure 2.17: **Conceptual Role of World Models** [Yan22]: world models integrate perception, memory, and action by both understanding the environment and simulating future dynamics, providing a generative backbone for prediction and planning.

AR training with self-refinement: instead of relying solely on teacher-forced ground-truth sequences, the model learns to generate from its own predictions, reducing exposure bias and improving long-horizon consistency. Figure 2.16 illustrates the conceptual differences between teacher forcing, diffusion forcing, and self-forcing.

2.7.4 Towards World Models

Autoregressive Transformers have recently been explored as *world models*, where the generative model is treated as a simulator of environment dynamics: given a stream of multimodal observations (e.g., images, actions, language), the model autoregressively predicts future states, as illustrated in Fig. 2.17. This extends AR beyond static generation, enabling the capture of temporal dynamics, causal dependencies, and latent world structure.

Beyond specific architectures, it is useful to distinguish the overarching *goals* and *approaches* of world models, as summarised in Fig. 2.18. At a high level, world models can either aim to *understand and predict the abstraction of the world* (e.g., JEPA) or to *generate and simulate the details of the world* (e.g., video world models, simulation platforms, neural 3D/4D). This perspective clarifies the

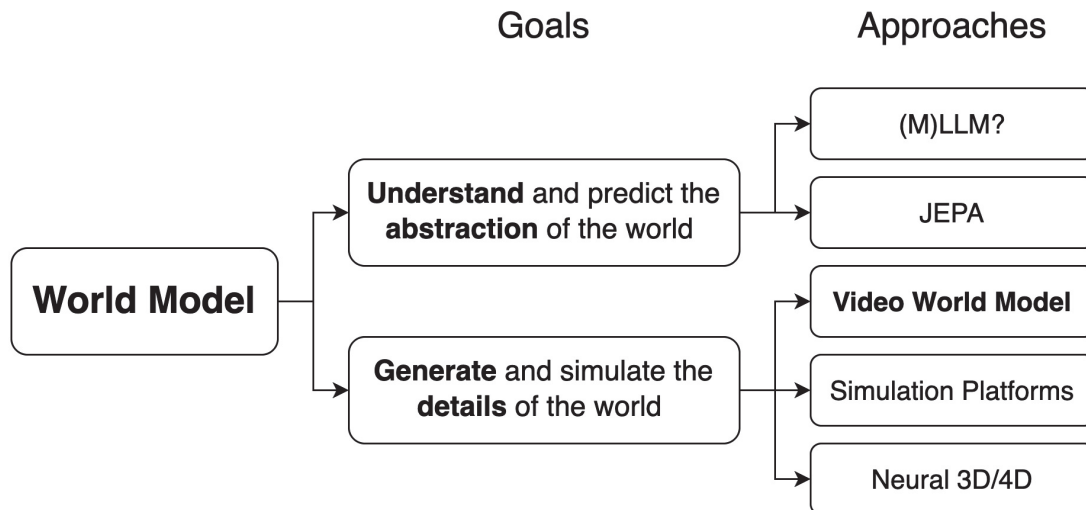


Figure 2.18: **Goals and Approaches of World Models** [Hua25]: World models can either focus on understanding and predicting abstractions of the world (e.g., LLMs, JEPA) or on generating and simulating world details (e.g., video world models, simulation platforms, neural 3D/4D).

dual role of world models: reasoning abstractly about latent structure versus producing concrete, detailed rollouts for interaction and perception.

Early efforts such as World Models [HS18] and Dreamer [HLBN20, HLN21, HPBL23] demonstrated that latent AR dynamics could support reinforcement learning by generating imagined trajectories for planning and control. More recent large-scale visual world models highlight the potential of AR Transformers for interactive environments. Genie [BDE⁺24, PHBB⁺24] tokenises videos into discrete representations and trains an AR Transformer to predict future frames, allowing agents to “imagine” gameplay from visual tokens alone. GameNGen [VLA25] builds on this idea by incorporating structural priors from game-like environments, achieving controllable and physically plausible rollouts conditioned on actions or rules.

A very recent milestone is Genie 3 [BBB⁺25], which pushes world models toward interactive, consistent, and realistic virtual environments. Genie 3 supports real-time environment generation (24 fps, 720p) from text prompts, allows agent navigation and interaction, and maintains temporal consistency over several minutes of rollout. These features mark a step change from earlier static or short-horizon video generators, moving closer to persistent and embodied world modelling. Nevertheless, many architectural details remain undisclosed, including how memory is represented, how

physical constraints are enforced, and how AR-style decoding interacts with latent rollouts.

Despite rapid progress, world models still face fundamental challenges. *Scalability* remains difficult, as long-horizon rollouts accumulate errors and require efficient memory or compression. *Consistency* across time and modality is non-trivial, since rollouts must remain physically plausible, visually coherent, and semantically stable. *Multimodality* increases complexity, as models must align images, actions, and language. Finally, *evaluation* of world models is still an open question, ranging from perceptual fidelity to task success. These open issues manifest differently depending on the goals in Fig. 2.18: models targeting *abstraction and prediction* (e.g., JEPA) must capture causal structure and latent dynamics, whereas models focused on *generation and simulation of details* (e.g., video world models, neural 3D/4D) must ensure realism, consistency, and controllability in long rollouts.

In this thesis, Chapter 5 connects to this perspective by introducing *CausNVS*, which integrates autoregressive conditioning with diffusion-based generation for multi-view synthesis. By applying causal masking across views, the model enforces consistent dependencies between reference and target images, serving as a step towards world-consistent generative modelling in 3D vision.

Chapter 3

vMAP: Vectorised Object-level Neural Mapping

Contents

3.1	Introduction	38
3.2	Related Work	39
3.3	vMAP: Object Mapping System with Vectorised Training	41
3.3.1	System Overview	41
3.3.2	Vectorised Object Level Mapping	41
3.3.3	Neural Implicit Mapping	44
3.3.4	Compositional Scene Rendering	46
3.4	Experiments	46
3.4.1	Experimental Setup	46
3.4.2	Evaluation on Scene and Object Reconstruction	49
3.4.3	Performance Analysis	56
3.4.4	Real-to-Sim-to-Real for Robot Learning	59
3.5	Conclusion	60



Figure 3.1: Overview of vMAP Reconstruction Results. vMAP automatically builds an object-level scene model from a real-time RGB-D input stream. Each object is represented by a separate MLP neural field model, all optimised in parallel via vectorised training. We use no 3D shape priors, but the MLP representation encourages object reconstruction to be watertight and complete, even when objects are partially observed or are heavily occluded in the input images. See for instance the separate reconstructions of the armchairs, sofas and cushions, which were mutually occluding each other, in this example from Replica.

In this chapter, we present vMAP, an object-level dense SLAM system using neural field representations. Each object is represented by a small MLP, enabling efficient, watertight object modelling without the need for 3D priors.

As an RGB-D camera browses a scene with no prior information, vMAP detects object instances on-the-fly, and dynamically adds them to its map. Specifically, thanks to the power of vectorised training, vMAP can optimise as many as 50 individual objects in a single scene, with an extremely efficient training speed of 5Hz map update. We experimentally demonstrate significantly improved scene-level and object-level reconstruction quality compared to prior neural field SLAM systems. Project page: <https://kxhit.github.io/vMAP>.

3.1 Introduction

For robotics and other interactive vision applications, an object-level model is arguably semantically optimal, with scene entities represented in a separated, composable way, but also efficiently focusing resources on what is important in an environment.

The key question in building an object-level mapping system is what level of prior information is known about the objects in a scene in order to segment, classify and reconstruct them. If no 3D object

priors are available, then usually only the directly observed parts of objects can be reconstructed, leading to holes and missing parts [ZK13, DNZ⁺17]. Prior object information such as CAD models or category-level shape space models enable full object shape estimation from partial views, but only for the subset of objects in a scene for which these models are available.

In this chapter, we present a new approach which applies to the case where no 3D priors are available but still often enables watertight object reconstruction in realistic real-time scene scanning. Our system, vMAP, builds on the attractive properties shown by neural fields as a real-time scene representation [SLOD21], with efficient and complete representation of shape, but now reconstructs a separate tiny MLP model of each object. The key technical contribution of our work is to show that a large number of separate MLP object models can be simultaneously and efficiently optimised on a single GPU during live operation via vectorised training.

We show that we can achieve much more accurate and complete scene reconstruction by separately modelling objects, compared with using a similar number of weights in a single neural field model of the whole scene. Our real-time system is highly efficient in terms of both computation and memory, and we show that scenes with up to 50 objects can be mapped with 40KB per object of learned parameters across the multiple, independent object networks.

We also demonstrate the flexibility of our disentangled object representation to enable recomposition of scenes with new object configurations. Extensive experiments have been conducted on both simulated and real-world datasets, showing state-of-the-art scene-level and object-level reconstruction performance.

3.2 Related Work

This work follows in long series of efforts to build real-time scene representations which are decomposed into explicit rigid objects, with the promise of flexible and efficient scene representation and even the possibility to represent changing scenes. Different systems assumed varying types of representation and levels of prior knowledge, from CAD models [SMNS⁺13], via category-level shape

models [SWD20, LLY⁺22, WRA21, KYZ⁺20] to no prior shape knowledge, although in this case only the visible parts of objects could be reconstructed [RA17b, MCB⁺18, XLT⁺19].

Neural Field SLAM Neural fields have recently been widely used as efficient, accurate and flexible representations of whole scenes [MST⁺20b, MESK22, MON⁺19, PFS⁺19]. To adopt these representations into real-time SLAM systems, iMAP [SLOD21] demonstrated for the first time that a simple MLP network, incrementally trained with the aid of depth measurements from RGB-D sensors, can represent room-scaled 3D scenes in real-time. Some of iMAP’s most interesting properties were its tendency to produce watertight reconstructions, even often plausibly completing the unobserved back of objects. These coherence properties of neural fields were particularly revealed when semantic output channels were added, as in SemanticNeRF [ZBLD19] and iLabel [ZSM⁺22], and were found to inherit the coherence. To make implicit representation more scalable and efficient, a group of implicit SLAM systems [ZPL⁺22, WBA22, YLZ⁺22, ZPBS23, RLC23] fused neural fields with conventional volumetric representations.

Object Representations with Neural Fields However, obtaining individual object representations from these neural field methods is difficult, as the correspondences between network parameters and specific scene regions are complicated and difficult to determine. To tackle this, DeRF [RJY⁺21] decomposed a scene spatially and dedicated smaller networks to each decomposed part. Similarly, KiloNeRF [RPLG21] divided a scene into thousands of volumetric parts, each represented by a tiny MLP, and trained them in parallel with custom CUDA kernels to speed up NeRF. Different from KiloNeRF, vMAP decomposes the scene into objects which are semantically meaningful.

To represent multiple objects, ObjectNeRF [YZX⁺21] and ObjSDF [WLC⁺22] took pre-computed instance masks as additional input and conditioned object representation on learnable object activation code. But these methods are still trained offline and tangle object representations with the main scene network, so that they need to optimise the network weights with all object codes during training, and infer the whole network to get the shape of a desired object. This contrasts with vMAP which models objects individually, and is able to stop and resume training for any objects without any inter-object interference.

The recent work most similar to ours has used the attractive properties of neural field MLPs to represent single objects. The analysis in [DNJ21] explicitly evaluated the use of over-fit neural implicit networks as a 3D shape representation for graphics, considering that they should be taken seriously. The work in [ACDS22] furthered this analysis, showing how object representation was affected by different observation conditions, though using the hybrid Instant NGP rather than a single MLP representation, so it is not clear whether some object coherence properties would be lost. Finally, the CodeNeRF system[JA21] trained a NeRF conditioned on learnable object codes, again proving the attractive properties of neural fields to represent single objects.

We build on this work in this chapter, but for the first time show that many individual neural field models making up a whole scene can be simultaneously trained within a real-time system, resulting in accurate and efficient representation of many-object scenes.

3.3 vMAP: Object Mapping System with Vectorised Training

3.3.1 System Overview

We first introduce our detailed design for object-level mapping with efficient vectorised training (Section 3.3.2), and then explain our improved training strategies of pixel sampling and surface rendering (Section 3.3.3). Finally, we show how we may recompose and render a new scene with these learned object models (Section 3.3.4). An overview of our training and rendering pipeline is shown in Fig. 3.2.

3.3.2 Vectorised Object Level Mapping

Object Initialisation and Association To start with, each frame is associated with densely labelled object masks. These object masks are either directly provided in the dataset, or predicted with an off-the-shelf 2D instance segmentation network. Since those predicted object masks have no temporal consistency across different frames, we perform object association between the previous and the current live frame, based on two criteria: i) *Semantic Consistency*: the object in the current frame is

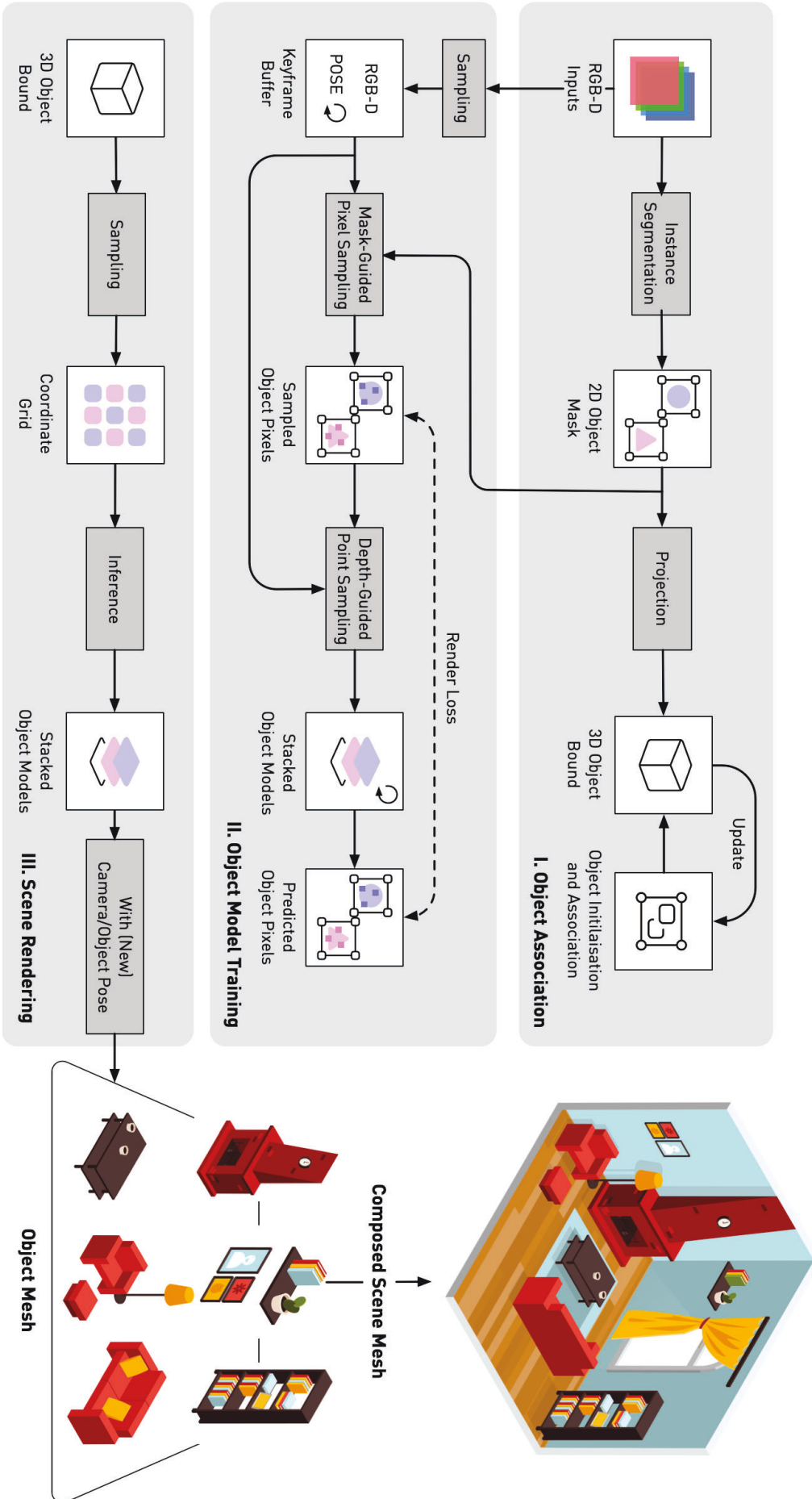


Figure 3.2: Overview of Training and Rendering Pipeline of vMAP.

predicted as the same semantic class from the previous frame, and ii) *Spatial Consistency*: the object in the current frame is spatially close to the object in the previous frames, measured by the mean IoU of their 3D object bounds. When these two criteria are satisfied, we assume they are the same object instance and represent them with the same object model. Otherwise, they are different object instances and we initialise a new object model and append it to the models stack.

For each object in a frame, we estimate its 3D object bound by its 3D point cloud, parameterised by its depth map and the camera pose. Camera tracking is externally provided by an off-the-shelf tracking system, which we found to be more accurate and robust compared to jointly optimising pose and geometry. If we detect the same object instance in a new frame, we merge its 3D point cloud from the previous frames to the current frame and re-estimate its 3D object bound. Therefore, these object bounds are dynamically updated and refined with more observations.

Object Supervision We apply object-level supervision only for pixels inside a 2D object bounding box, for maximal training efficiency. For those pixels within an object mask, we encourage the object radiance field to be occupied and supervise them with depth and colour loss. Otherwise we encourage the object radiance field to be empty.

Each object instance samples training pixels from its own independent keyframe buffer. Therefore, we have flexibility to stop or resume the training of any object, with no training interference between objects.

Vectorised Training Representing a neural field with multiple small networks can lead to efficient training, as shown in prior work [RPLG21]. In vMAP, all object models are of the same design, except for the background object which we represent with a slightly larger network. Therefore, we are able to stack these small object models together for vectorised training, leveraging the highly optimised vectorised operations in PyTorch [HR21]. Since multiple object models are batched and trained simultaneously as opposed to sequentially, we optimise the use of the available GPU resources. We show that vectorised training is an essential design element to the system, resulting in significantly improved training speed, further discussed in Section 3.4.3.

3.3.3 Neural Implicit Mapping

Depth Guided Sampling Neural fields trained on RGB data only have no guarantee to model accurate object geometry, due to the fact that they are optimising for appearance rather than the geometry. To obtain more geometrically accurate object models, we benefit from the depth map available from an RGB-D sensor, providing a strong prior for learning the density field of 3D volumes. Specifically, we sample N_s and N_c points along each ray, for which N_s points are sampled with a Normal distribution centered around the surface t_s (from the depth map), with a small d_σ variance, and N_c points are uniformly sampled between the camera t_n (the near bound) and the surface t_s , with a stratified sampling approach. When the depth measurement is invalid, the surface t_s is then replaced with the far bound t_f . Mathematically, we have:

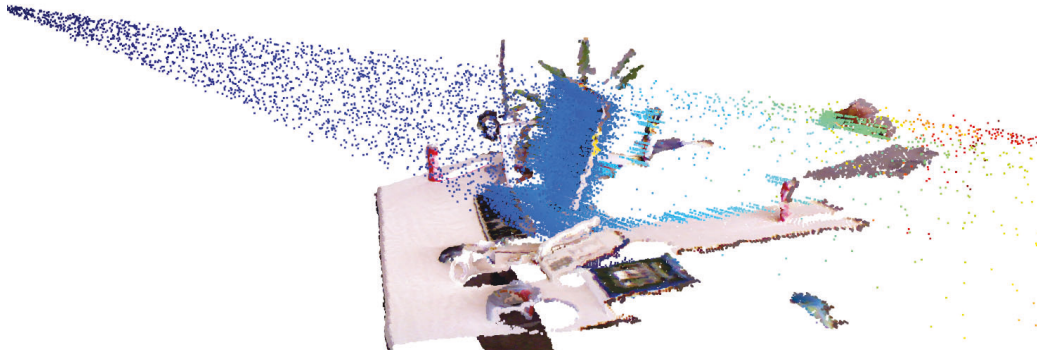
$$t_i \sim \mathcal{U} \left(t_n + \frac{i-1}{N_c} (t_s - t_n), t_n + \frac{i}{N_c} (t_s - t_n) \right), \quad (3.1)$$

$$t_i \sim \mathcal{N}(t_s, d_\sigma^2). \quad (3.2)$$

We choose $d_\sigma = 3cm$ which works well in our implementation. We observe that training more points near the surface helps to guide the object models to quickly focus on representing accurate object geometry.

As illustrated in Fig. 3.3 with sampled points coloured by the measured depth, we sampled more points near the object surface guided by the depth measurements. For rays that go through the 3D object bounding box but do not belong to the current instance, we then terminate these rays when they hit the object surface, to minimise the impact on the occluded objects, similar to ObjectNeRF [YZX⁺21].

Surface and Volume Rendering As we are concerned more by 3D surface reconstruction than 2D rendering, we omit the viewing direction from the network input, and model object visibility with a binary indicator (no transparent objects). With similar motivation to UniSURF [OPG21], we parameterise the occupancy probability of a 3D point x_i as $o_\theta(x_i) \rightarrow [0, 1]$, where o_θ is a continuous occupancy field. Therefore, the termination probability at point x_i along ray $\mathbf{r}$ becomes $T_i = o(x_i) \prod_{j < i} (1 - o(x_j))$, indicating that no occupied samples x_j with $j < i$ exist before x_i . The

Figure 3.3: **Visualisation of Depth-guided Sampling.**

corresponding rendered occupancy, depth and colour are defined as follows:

$$\hat{O}(\mathbf{r}) = \sum_{i=1}^N T_i, \quad \hat{D}(\mathbf{r}) = \sum_{i=1}^N T_i d_i, \quad \hat{C}(\mathbf{r}) = \sum_{i=1}^N T_i c_i. \quad (3.3)$$

Training Objective For each object k , we only sample training pixels inside that object’s 2D bounding box, denoted by $\mathcal{R}^k$, and only optimise depth and colour for pixels inside its 2D object mask, denoted by M^k . Note that it is always true that $M^k \subset \mathcal{R}^k$. The depth, colour and occupancy loss for the object k are defined as follows:

$$L_{depth}^k = M^k \odot \sum_{\mathbf{r} \in \mathcal{R}^k} |\hat{D}(\mathbf{r}) - D(\mathbf{r})|, \quad (3.4)$$

$$L_{colour}^k = M^k \odot \sum_{\mathbf{r} \in \mathcal{R}^k} |\hat{C}(\mathbf{r}) - C(\mathbf{r})|, \quad (3.5)$$

$$L_{occupancy}^k = \sum_{\mathbf{r} \in \mathcal{R}^k} |\hat{O}(\mathbf{r}) - M^k(\mathbf{r})|. \quad (3.6)$$

The overall training objective then accumulates losses for all K objects:

$$L = \sum_{k=1}^K L_{depth}^k + \lambda_1 \cdot L_{colour}^k + \lambda_2 \cdot L_{occupancy}^k. \quad (3.7)$$

We choose loss weightings $\lambda_1 = 5$ and $\lambda_2 = 10$, which we found to work well in our experiments.

3.3.4 Compositional Scene Rendering

Since vMAP represents objects in a purely disentangled representation space, we can obtain each 3D object by querying within its estimated 3D object bounds and easily manipulate it. For 2D novel view synthesis, we use the Ray-Box Intersection algorithm [MCSM18] to calculate near and far bounds for each object, and then rank rendered depths along each ray to achieve occlusion-aware scene-level rendering. This disentangled representation also opens up other types of fine-grained object-level manipulation, such as changing object shape or textures by conditioning on disentangled pre-trained feature fields [NG21, YSL⁺22], which we consider as an interesting future direction.

3.4 Experiments

We have comprehensively evaluated vMAP on a range of different datasets, which include both simulated and real-world sequences, with and without ground-truth object masks and poses. For all datasets, we qualitatively compare our system to prior state-of-the-art SLAM frameworks on 2D and 3D scene-level and object-level rendering. We further quantitatively compare these systems in datasets where ground-truth meshes are available.

3.4.1 Experimental Setup

Datasets We evaluated on Replica [SWM⁺19], ScanNet [DCS⁺17], and TUM RGB-D [EHE⁺12]. Each dataset contains sequences with different levels of quality in object masks, depth and pose measurements. Additionally, we also showed vMAP’s performance in complex real-world with self-captured video sequences recorded by an Azure Kinect RGB-D camera. An overview of these datasets is shown in Tab. 3.1.

Datasets with perfect ground-truth information represent the upper-bound performance of our system. We expect vMAP’s performance in the real-world setting can be further improved, when coupled with a better instance segmentation and pose estimation framework.

	Object Masks	Depth Quality	Pose Estimation
Replica	Perfect GT	Perfect GT	Perfect GT
ScanNet	Noisy	Noisy	Perfect GT
TUM RGB-D	Detic	Noisy	ORB-SLAM3
Our Recording	Detic	Noisy	ORB-SLAM3

Table 3.1: Overview of Evaluated Datasets.

Metric	TSDF-Fusion*	iMAP	iMAP*	NICE-SLAM	NICE-SLAM*	Ours
Scene Acc. [cm] ↓	1.28	4.43	2.15	2.94	3.04	3.20
Scene Comp. [cm] ↓	5.61	5.56	2.88	4.02	3.84	2.39
Scene Comp. Ratio [$<5\text{cm}$ %] ↑	82.67	79.06	90.85	86.73	86.52	92.99
Object Acc. [cm] ↓	0.45	–	3.57	–	3.91	2.23
Object Comp. [cm] ↓	3.69	–	2.38	–	3.27	1.44
Object Comp. Ratio [$<5\text{cm}$ %] ↑	82.98	–	90.19	–	83.97	94.55
Object Comp. Ratio [$<1\text{cm}$ %] ↑	61.70	–	47.79	–	37.79	69.23

Table 3.2: Reconstruction Results for 8 Indoor Replica Scenes. * indicates methods re-trained with ground-truth poses.

Implementation Details We conduct all experiments on a desktop PC with a 3.60 GHz i7-11700K CPU and a single Nvidia RTX 3090 GPU. We choose our instance segmentation detector to be Detic [ZGJ⁺22], pre-trained on an open-vocabulary LVIS dataset [GDG19] which contains more than 1000 object classes. We choose our pose estimation framework to be ORB-SLAM3 [CER⁺21], for its fast and accurate tracking performance. We continuously update the keyframe poses using the latest estimates from ORB-SLAM3.

We applied the same set of hyper-parameters for all datasets. Both our object and background model use 4-layer MLPs, with each layer having hidden size 32 (object) and 128 (background). For object / background, we selected keyframes every 25 / 50 frames, 120 / 1200 rays each training step with 10 points per ray. The number of objects in a scene typically varies between 20 and 70, among which the largest number of objects are in Replica and ScanNet scenes with an average of 50 objects per scene.

Metrics Following the convention of prior work [SLOD21, ZPL⁺22], we adopt *Accuracy*, *Completion*, and *Completion Ratio* for 3D scene-level reconstruction metrics. Besides, we note that such

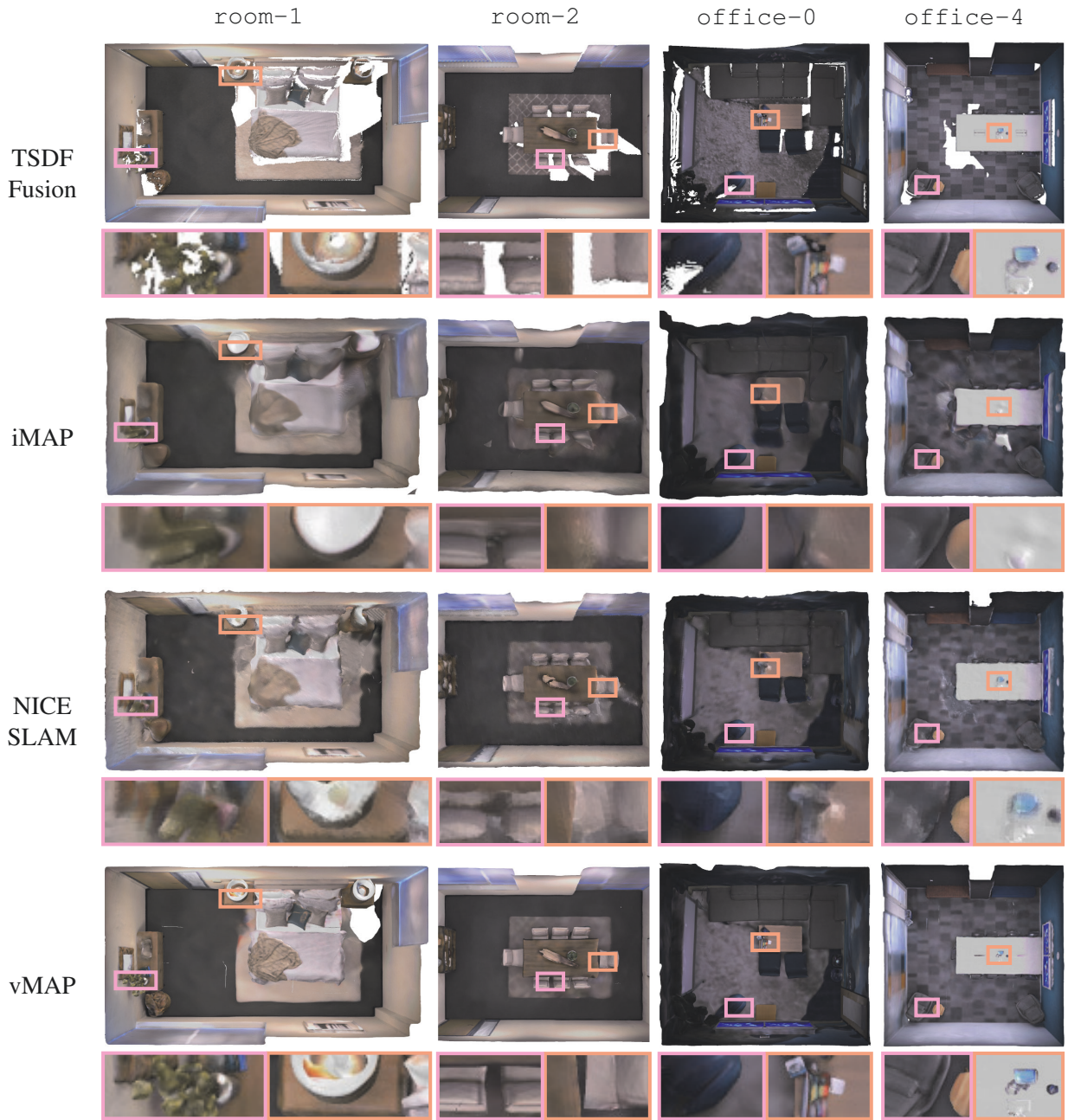


Figure 3.4: **Scene Reconstruction Visualisation for 4 Selected Replica Scenes.** Interesting regions are highlighted with coloured boxes, showing vMAP’s significantly improved reconstruction quality. All scene meshes are provided by the original authors.

scene-level metrics are heavily biased towards the reconstruction of large objects like walls and floors. Therefore, we additionally provide these metrics at the object-level, by averaging metrics for all objects in each scene.

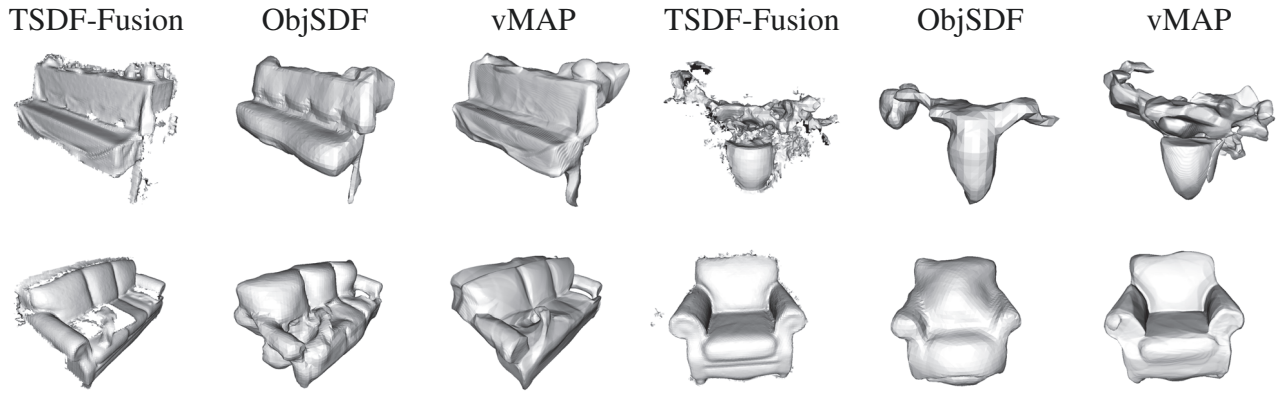


Figure 3.5: **Visual Comparison at Object-level Reconstruction.** Visualisation of object reconstructions with vMAP compared to TSDF-Fusion and ObjSDF. Note that all object reconstructions from ObjSDF require much longer off-line training. All object meshes from ObjSDF are provided by the original authors.



Figure 3.6: **Reconstruction Visualisation on ScanNet Dataset.** Visualisation of scene reconstruction from NICE-SLAM* (left) and vMAP (right) in a selected ScanNet sequence. Interesting regions are zoomed in. NICE-SLAM* was re-trained with ground-truth poses.

3.4.2 Evaluation on Scene and Object Reconstruction

Results on Replica We experimented on 8 Replica scenes, using the rendered trajectories provided in [SLOD21], with 2000 RGB-D frames in each scene. Tab. 3.2 shows the averaged quantitative reconstruction results in these Replica indoor sequences. For scene-level reconstruction, we compared with TSDF-Fusion [ZK13], iMAP [SLOD21] and NICE-SLAM [ZPL⁺22]. To isolate reconstruction, we also provided results for these baselines re-trained with ground-truth pose (marked with *), with their open-sourced code for the fair comparison. Specifically, iMAP* was implemented as a special case of vMAP, when considering the entire scene as one object instance. For object-level reconstruction, we compared baselines trained with ground-truth pose.

vMAP’s significant advantage thanks to object-level representation is to reconstruct tiny objects and

		<i>room-0</i>	<i>room-1</i>	<i>room-2</i>	<i>office-0</i>	<i>office-1</i>	<i>office-2</i>	<i>office-3</i>	<i>office-4</i>	<i>Avg.</i>
TSDF-Fusion*	Acc. [cm] ↓	1.46	1.13	1.22	1.13	0.91	1.33	1.56	1.48	1.28
	Comp. [cm]	3.73	3.51	4.41	10.26	9.57	5.50	3.87	4.04	5.61
	Ratio [$<5\text{cm}$ %]	86.54	87.12	84.87	78.86	75.85	80.48	83.19	84.41	82.67
iMAP	Acc. [cm] ↓	3.58	3.69	4.68	5.87	3.71	4.81	4.27	4.83	4.43
	Comp. [cm] ↓	5.06	4.87	5.51	6.11	5.26	5.65	5.45	6.59	5.56
	Ratio [$<5\text{cm}$ %] ↑	83.91	83.45	75.53	77.71	79.64	77.22	77.34	77.63	79.06
iMAP*	Acc. [cm] ↓	2.06	1.65	1.92	2.36	1.94	2.61	2.41	2.23	2.15
	Comp. [cm] ↓	2.21	1.94	2.77	4.81	3.19	2.81	2.78	2.56	2.88
	Ratio [$<5\text{cm}$ %] ↑	94.93	94.87	90.78	86.85	87.79	89.61	90.54	91.46	90.85
NICE-SLAM	Acc. [cm] ↓	2.69	2.49	2.55	3.03	3.31	3.56	3.26	2.63	2.94
	Comp. [cm] ↓	2.92	2.33	2.96	8.34	5.18	3.35	3.37	3.68	4.02
	Ratio [$<5\text{cm}$ %] ↑	90.77	93.07	87.83	81.99	82.24	85.82	85.44	86.64	86.73
NICE-SLAM*	Acc. [cm] ↓	2.71	2.28	2.69	2.93	4.23	3.45	3.26	2.74	3.04
	Comp. [cm] ↓	2.84	2.23	3.02	7.54	4.52	3.31	3.58	3.64	3.84
	Ratio [$<5\text{cm}$ %] ↑	91.00	93.37	87.23	82.70	82.09	85.42	84.28	86.10	86.52
vMAP	Acc. [cm] ↓	2.77	3.87	1.83	4.82	3.51	3.35	3.19	2.26	3.20
	Comp. [cm] ↓	1.99	1.81	2.00	3.65	2.14	2.45	2.49	2.56	2.39
	Ratio [$<5\text{cm}$ %] ↑	97.10	96.59	95.72	87.53	85.08	94.70	93.65	93.56	92.99

Table 3.3: **Scene-level Reconstruction Results on 8 Indoor Replica Scenes.** * denotes re-training with ground-truth pose.

objects with fine-grained details. Noticeably, vMAP achieved more than 50 – 70% improvement over iMAP and NICE-SLAM for object-level completion. The scene reconstructions of 4 selected Replica sequences are shown in Fig. 3.4, with interesting regions highlighted in coloured boxes.

In Tab. 3.3 and Tab. 3.4, we listed the detailed scene-level and object-level 3D reconstruction results for each sequence on Replica dataset.

We generated a new / different sequence for each scene in Replica dataset. We performed 2D novel view synthesis and compared it to the ground-truth views from the generated sequence. We compared baselines in depth L1 error, PSNR, SSIM, and LPIPS in Tab. 3.5, and the 2D renderings for 3 selected scenes are shown in Fig. 3.7.

Results on ScanNet To evaluate on a more challenging setting, we experimented on ScanNet, a dataset composed of real scenes [DCS⁺17], with much noisier ground-truth depth maps and object masks. We choose a ScanNet sequence selected by ObjSDF [WLC⁺22], and we compared with TSDF-Fusion and ObjSDF for object-level reconstruction, and we compared with NICE-SLAM (re-

		room-0	room-1	room-2	office-0	office-1	office-2	office-3	office-4	Avg.
TSDF-Fusion*	Acc. [cm] ↓	0.43	0.45	0.45	0.49	0.43	0.41	0.45	0.52	0.45
	Comp. [cm] ↓	3.03	4.42	4.16	2.23	5.60	3.32	3.31	3.48	3.69
	Ratio [<1cm %] ↑	62.27	59.44	53.57	68.16	55.73	67.34	63.35	63.75	61.70
	Ratio [<5cm %] ↑	84.34	79.12	76.89	86.74	80.36	87.30	85.74	83.38	82.98
NICE-SLAM*	Acc. [cm] ↓	3.48	3.77	4.61	4.08	3.42	3.45	3.96	4.53	3.91
	Comp. [cm] ↓	2.51	2.82	3.19	3.05	3.29	3.47	3.61	4.23	3.27
	Ratio [<1cm %] ↑	41.19	37.06	33.03	38.86	44.55	41.84	31.21	34.54	37.79
	Ratio [<5cm %] ↑	86.88	86.43	83.96	84.54	89.08	83.77	79.40	77.64	83.96
iMAP*	Acc. [cm] ↓	3.02	3.35	4.50	3.84	2.62	3.22	3.58	4.43	3.57
	Comp. [cm] ↓	1.71	1.93	3.45	1.66	2.58	2.28	2.32	3.14	2.38
	Ratio [<1cm %] ↑	52.57	43.56	45.06	48.16	48.93	53.59	51.07	39.36	47.79
	Ratio [<5cm %] ↑	93.72	92.95	85.30	94.56	91.09	89.99	89.32	84.60	90.19
vMAP	Acc. [cm] ↓	2.18	3.46	2.01	2.37	2.27	1.75	1.90	1.93	2.23
	Comp. [cm] ↓	1.13	1.54	1.58	1.15	1.77	1.03	1.42	1.94	1.44
	Ratio [<1cm %] ↑	74.09	68.51	66.81	67.00	65.24	77.98	68.62	65.56	69.23
	Ratio [<5cm %] ↑	96.68	95.02	92.98	96.53	92.94	96.97	94.21	91.03	94.55

Table 3.4: **Object-level Reconstruction Results on 8 Indoor Replica Scenes.** * denotes re-training with ground-truth pose.

		room-0	room-1	room-2	office-0	office-1	office-2	office-3	office-4	Avg.
NICE-SLAM	Depth L1 [cm] ↓	1.99	1.57	2.72	12.50	7.37	3.03	2.39	2.18	4.22
	PSNR ↑	24.11	23.43	23.48	23.91	22.69	23.78	23.78	26.00	23.90
	SSIM ↑	0.73	0.74	0.82	0.83	0.82	0.83	0.84	0.85	0.81
	LPIPS ↓	0.11	0.09	0.09	0.15	0.28	0.11	0.10	0.09	0.13
NICE-SLAM*	Depth L1 [cm] ↓	1.87	1.63	2.94	13.43	7.63	2.83	2.62	1.97	4.36
	PSNR ↑	24.03	23.61	23.54	23.59	23.19	22.22	23.32	26.20	23.71
	SSIM ↑	0.73	0.75	0.82	0.83	0.84	0.85	0.84	0.86	0.82
	LPIPS ↓	0.11	0.09	0.09	0.16	0.26	0.10	0.10	0.09	0.13
iMAP*	Depth L1 [cm] ↓	1.23	2.16	2.53	13.29	5.14	2.31	1.77	1.44	3.73
	PSNR ↑	25.83	25.51	25.22	24.17	23.94	24.02	25.45	29.13	25.41
	SSIM ↑	0.77	0.79	0.86	0.83	0.87	0.88	0.89	0.90	0.85
	LPIPS ↓	0.09	0.07	0.07	0.17	0.22	0.08	0.07	0.07	0.11
vMAP	Depth L1 [cm] ↓	1.68	1.57	2.37	7.73	6.60	2.50	2.30	1.85	3.33
	PSNR ↑	25.23	25.27	24.31	23.78	23.59	23.10	23.83	27.91	24.63
	SSIM ↑	0.77	0.78	0.85	0.84	0.88	0.88	0.88	0.89	0.85
	LPIPS ↓	0.09	0.07	0.08	0.16	0.23	0.07	0.08	0.07	0.11

Table 3.5: **2D Novel View Synthesis Rendering Results on Replica Dataset.**

trained with ground-truth pose) for scene-level reconstruction. Unlike ObjSDF, which was optimised from pre-selected posed images without depth for much longer off-line training, we ran both vMAP and TSDF-Fusion in an online setting with depth. As shown in Fig. 3.5, we see that vMAP generates

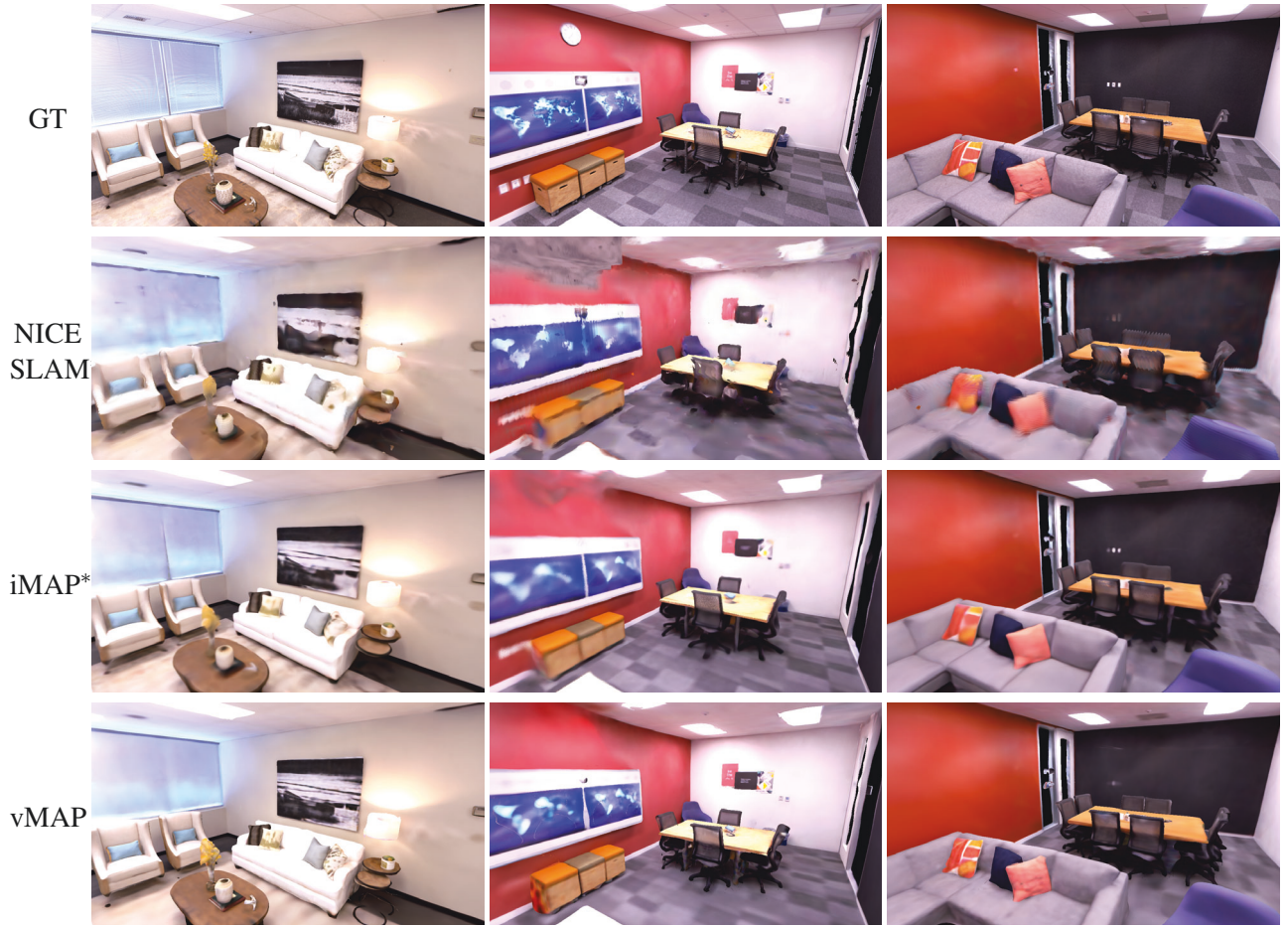


Figure 3.7: **Visualisation of 2D Novel View Synthesis on Replica Dataset.** Better when zoomed.

objects with more coherent geometry than TSDF-Fusion; and with much finer details than ObjSDF, though with a much shorter training time. And consistently, we can see that vMAP generates much sharper object boundaries and textures compared to NICE-SLAM, as shown in Fig. 3.6.

Results on TUM RGB-D We evaluated on a TUM RGB-D sequence captured in the real-world, with object masks predicted by an off-the-shelf pre-trained instance segmentation network [ZGJ⁺22], and poses estimated by ORB-SLAM3 [CER⁺21]. Since our object detector has no spatio-temporal consistency, we found that the same object can be occasionally detected as two different instances, which leads to some reconstruction artifacts. For example, the object ‘globe’ shown in Fig. 3.8 was also detected as ‘balloon’ in some frames, resulting the ‘splitting’ artifacts in the final object reconstruction. Overall, vMAP still predicts more coherent reconstruction for most objects in a scene, with realistic hole-filling capabilities compared to TSDF-Fusion. However, we acknowledge that the completion of complete out-of-view regions (e.g., the back of a chair) is beyond the reach of our



Figure 3.8: **Visualisation of Reconstruction on TUM RGB-D Dataset.** Visualisation of scene reconstruction from TSDf-Fusion (left) and vMAP (right) in a selected TUM RGB-D sequence, trained in real time for 99 seconds.

ATE RMSE [cm]↓	iMAP	NICE-SLAM	vMAP	ORB-SLAM2
fr1/desk	4.9	2.7	2.6	1.6
fr2/xyz	2.0	1.8	1.6	0.4
fr3/office	5.8	3.0	3.0	1.0

Table 3.6: **Camera Tracking Results on TUM RGB-D Dataset.**

system due to the lack of general 3D prior.

Though our work focuses more on mapping performance than pose estimation, we also report *ATE RMSE* [SEE⁺12] in Tab. 3.6 following [SLOD21, ZPL⁺22], by jointly optimising camera pose with map. We can observe that vMAP achieves superior performance, due to the fact that reconstruction and tracking quality are typically highly interdependent. However, there is a noticeable performance gap compared to ORB-SLAM. As such, we directly choose ORB-SLAM as our external tracking system, which leads to faster training speed, cleaner implementation, and higher tracking quality.

Results on Live Kinect Data Finally, we show the reconstruction result of vMAP on a table-top scene, from running in real-time with an Azure Kinect RGB-D camera. As shown in Fig. 3.9 and Fig. 3.10, vMAP is able to reconstruct a range of realistic, watertight object meshes from different categories in table-top and scene-level settings.



Figure 3.9: **Real-world Live Reconstruction (Table-top)**. Visualisation of table-top reconstruction (top) and individual object reconstructions (bottom), from vMAP running in real time using an Azure Kinect RGB-D camera for 170 seconds.

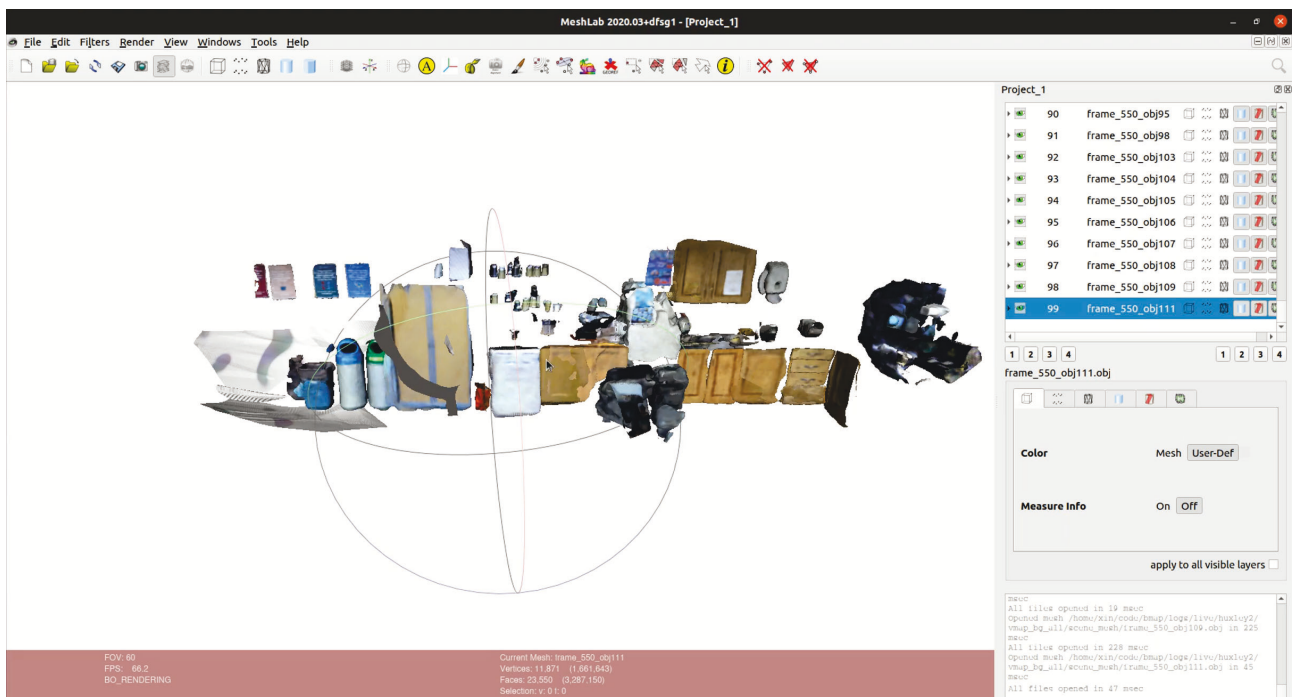


Figure 3.10: **Real-world Live Reconstruction (Scene-level)**. Visualisation of vMAP’s object-level reconstruction in an indoor scene at Huxley Common Room in Imperial College London.

Object-Level Positional Encoding Since object instances are different in size, the reconstruction quality can be maximised when trained with a suitable positional encoding frequency. Otherwise, the

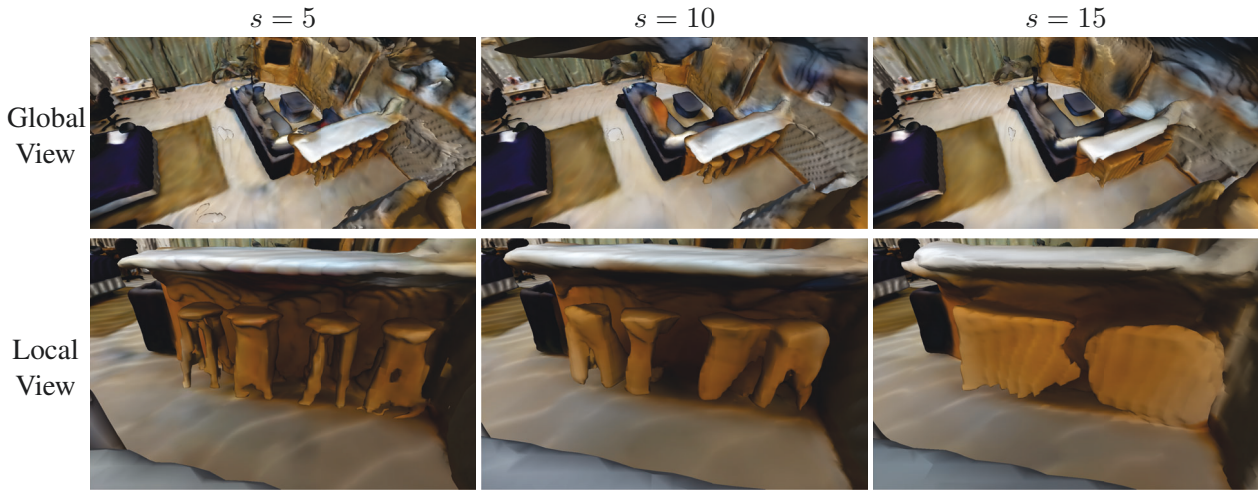


Figure 3.11: **Visualisation of Object Reconstructions Trained with Different Scales.**

network training would be biased towards reconstructing large objects and overlook small objects or vice versa. To mitigate this scaling issue, we applied integrated positional encoding [BMV⁺22] and introduced an additional hyper-parameter, the scaling factor s , which is applied to all objects, such that they are bounded in a unit box within the range of $[-1, 1]$. We separately set this scaling factor slightly larger in the background model.

This scaling factor can be set as object specific if such object-specific prior is known, i.e. we can set a large s when training the object ‘sofa’, and a small s when training the object ‘cup’, because a sofa is typically larger than a cup. A visualisation of the object reconstruction with different choices of s is shown in Fig. 3.11. We can see a large scale s results a smoother geometry which is more suitable for reconstructing large objects like ‘walls’ and ‘blankets’, and a small s is more suitable for objects with complex geometries like ‘chairs’.

Visualisation of Object-level Hole-filling Compared to iMAP and NICE-SLAM, vMAP shows significantly better hole-filling capability in unobserved regions with visual consistency, thanks to the disentangled object representation design. As shown in Fig. 3.12, vMAP is able to generate smooth and natural geometries without requiring any other priors.

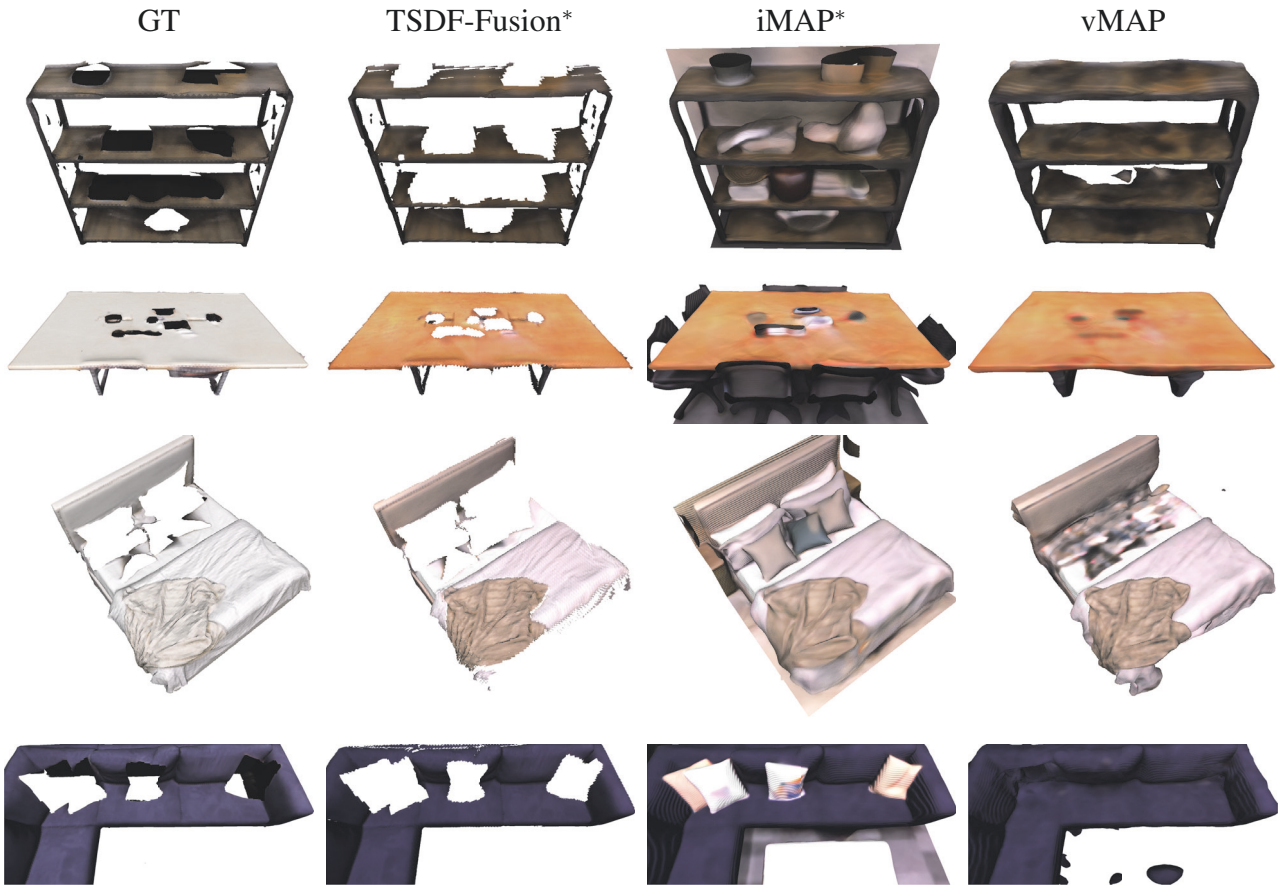


Figure 3.12: Visualisation of Object-level Hole-filling.

3.4.3 Performance Analysis

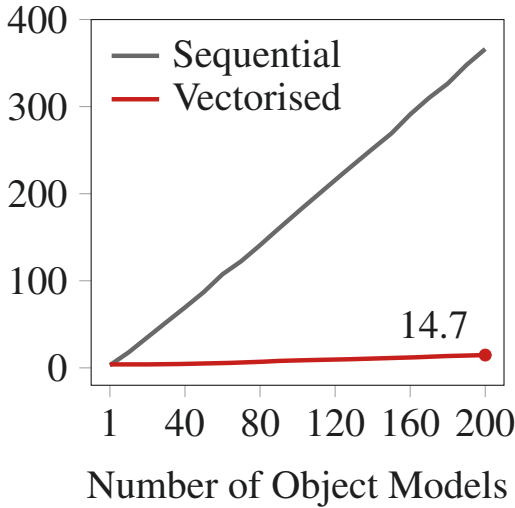
In this section, we compare different training strategies and architectural design choices for our vMAP system. For simplicity, all experiments were done on the Replica Room-0 sequence, with our default training hyper-parameters.

Memory and Runtime We compared memory usage and runtime with iMAP and NICE-SLAM in Tab. 3.7 and Fig. 3.14, all trained with ground-truth pose, and with the default training hyper-parameters listed in each method, for fair comparison. Specifically, we reported the *Runtime* for training the entire sequence, and *Mapping Time* for training each single frame, given the exact same hardware. We can observe that vMAP is highly memory efficient with less than 1M parameters. We want to highlight that vMAP achieves better reconstruction quality, and runs significantly faster ($\sim 5\text{Hz}$) than iMAP and NICE-SLAM with 1.5x and 4x training speed improvement respectively.

	NICE-SLAM*	iMAP	vMAP	vMAP (w/o BG)
Model Param. ↓	12.12M	0.32M	0.66M	0.56M
Runtime ↓	34min34s	12min29s	8min16s	6min01s
Mapping Time ↓	845ms	360ms	226ms	120ms

Table 3.7: **Memory and Runtime Efficiency.** vMAP is extremely memory-efficient and runs 1.5x and 4x faster than iMAP and NICE-SLAM respectively, with even higher performance gains without the background (BG) model.

Step Time (ms)



Step Time (ms)

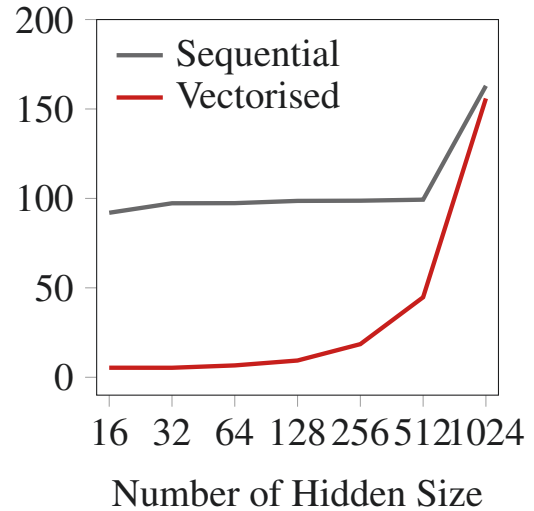


Figure 3.13: **Vectorised v.s. Sequential Training.** Vectorised operation allows extremely fast training speed compared to standard sequential operations using for loops.

Vectorised v.s. Sequential Training We ablated training speed with vectorised and sequential operations (for loops), conditioned on different numbers of objects and different sizes of object model. In Fig. 3.13, we can see that vectorised training enables tremendous improvements in optimisation speed, especially when we have a large number of objects. And with vectorised training, each optimisation step takes no more than 15ms even when we train as many as 200 objects. Additionally, vectorised training is also stable across a wide range of model sizes, suggesting that we can train our object models with an even larger size if required, with minimal additional training time. As expected, vectorised training and for loops will eventually have similar training speed, when we reach the hardware’s memory limit.

To train multiple models in parallel, an initial approach we tried was spawning a process per object. However, we were only able to spawn a very limited number of processes, due to the per process CUDA memory overhead, which significantly limited the number of objects.

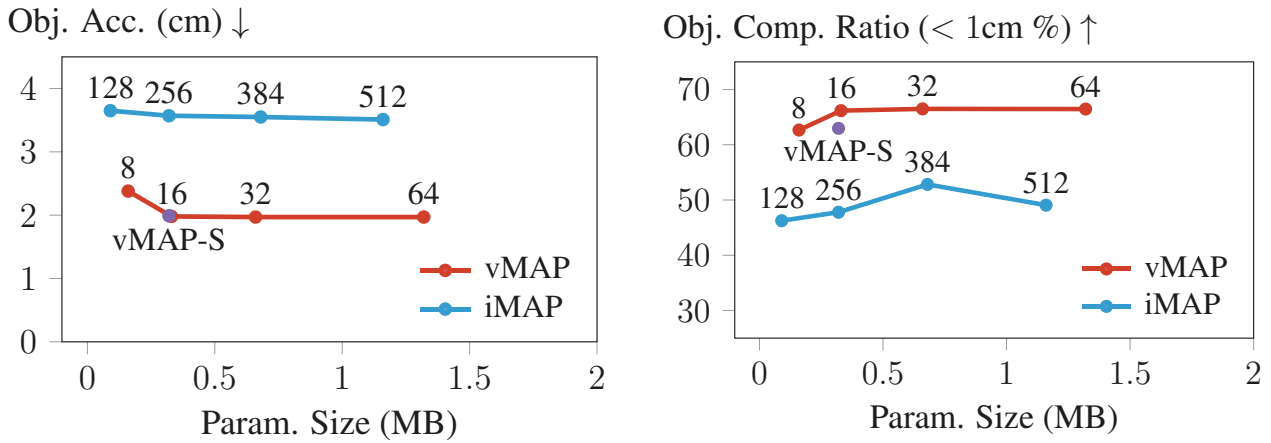


Figure 3.14: **Object-level Reconstruction Performance vs. Model Size.** The model parameter size is represented via hidden size. vMAP is more compact than iMAP, with performance saturating from hidden size 16.

Object Model Capacity As vectorised training has minimal effect on training speed in terms of object model design, we also investigated how the object-level reconstruction quality is affected by different object model sizes. We experimented with different object model sizes by varying the hidden size of each MLP layer. In Fig. 3.14, we can see that the object-level performance starts to saturate starting from hidden size 16, with minimal or no improvement by further increasing model sizes. This indicates that object-level representation is highly compressible, and can be efficiently and accurately parameterised by very few parameters.

Stacked MLPs v.s. Shared MLP Apart from representing each object by a single individual MLP, we also explored a shared MLP design by considering multi-object mapping as a multi-task learning problem [VGVG⁺21, Rud17]. Here, each object is additionally associated with a learnable latent code, and this latent code is considered as an conditional input to the network, jointly optimised with the network weights. Though we have tried multiple multi-task learning architectures [MSGH16, LJD19], early experiments (denoted as vMAP-S in Fig. 3.14) showed that this shared MLP design achieved slightly degraded reconstruction quality and had no distinct training speed improvement compared to stacked MLPs, particularly when powered by vectorised training. Furthermore, we found that shared MLP design can lead to undesired training properties: i) The shared MLP needs to be optimised along with the latent codes from all the objects, since the network weights and all object codes are *entangled* in a shared representation space. ii) The shared MLP capacity is *fixed* during

training, and therefore the representation space might not be sufficient with an increasing number of objects. This accentuates the advantages of disentangled object representation space, which is a crucial design element of vMAP system.

3.4.4 Real-to-Sim-to-Real for Robot Learning

A common direction in robot learning is to perform reinforcement learning in simulation with randomized environments, and then deploy the trained policy in the real world [HLD⁺19]. Our object-level reconstruction system can facilitate this process by creating digital twins, enabling the real world to be transferred into simulation for ultra-efficient parallel training.

To validate the reconstruction capability of vMAP in robotic manipulation tasks, we convert the object neural field into meshes using marching cubes [LC98], and assign affordance and physical parameters of both objects and environments in Isaac Gym [MWG⁺21], as shown in Fig. 3.15. Thanks to the watertight property of neural fields, the reconstructions can directly support physical simulation. However, we observe that the initial states of objects are sensitive to the reconstruction accuracy of the contact surfaces (e.g., the bottom of an object in contact with a table). Without incorporating priors in reconstruction or environment modeling, even small errors may lead to changes in these initial states. While this does not significantly affect policy training for interacting with a variety of objects, it poses challenges for maintaining an identical copy of the real world within a physics engine.

Importantly, this demonstrates a capability beyond both scene-level neural reconstructions without object decomposition (e.g., iMAP [SLOD21] and traditional object-level methods using TSDF (e.g, Fusion++ [MCB⁺18])), which often leave holes in unobserved regions. By producing watertight, simulation-ready object reconstructions, vMAP enables a closer alignment between real and simulated environments.



Figure 3.15: Load vMAP Reconstruction in Issac Gym Robotic Simulation.

3.5 Conclusion

We have presented vMAP, a real-time object-level mapping system with simple and compact neural implicit representation. By decomposing the 3D scene into meaningful instances, represented by a batch of tiny separate MLPs, the system models the 3D scene in an efficient and flexible way, enabling scene re-composition, independent tracking and continually updating of objects of interest. In addition to more accurate and compact object-centric 3D reconstruction, our system is able to predict plausible watertight surfaces for each object, even under partial occlusion.

Limitations and Future Work Our current system relies on an off-the-shelf detector for instance masks, which are not necessarily spatio-temporally consistent. Though the ambiguity is partially alleviated by data association and multi-view supervision, a reasonable global constraints will be better. As objects are modelled independently, dynamic objects can be continually tracked and reconstructed to enable downstream tasks, e.g., robotic manipulation[WSJ⁺20]. To extend our system to a monocular dense mapping system, depth estimation networks[YPN⁺22, LLW⁺21] or more efficient neural rendering approaches [MESK22] could be further integrated.

In this chapter, we have demonstrated how neural fields can be used for efficient object-level 3D scene reconstruction without relying on any 3D priors. While our approach is both compact and composable, it still requires dense observations and scene-specific optimization. In the next chapter, we move beyond per-scene modeling and investigate how to learn generalizable 3D priors through large-scale pre-training. By doing so, we aim to reconstruct objects and scenes from sparse observations, enabling a new level of scalability and generalization in 3D neural scene representations.

Chapter 4

EscherNet: Scalable 3D Multi-view Generative Synthesis

Contents

4.1	Introduction	64
4.2	Related Work	66
4.3	EscherNet	68
4.3.1	Architecture Design	70
4.3.2	Camera Positional Encoding (CaPE)	71
4.4	Experiments	76
4.4.1	Implementation Details	76
4.4.2	Evaluation Details	77
4.4.3	Results on Novel View Synthesis	78
4.4.4	Results on 3D Generation	81
4.4.5	Results on Text-to-3D Generation	83
4.4.6	Results with 6 DoF CaPE	85
4.5	Discussions	87
4.6	Conclusions	89



Figure 4.1: **Multi-view to Multi Novel View Synthesis.** We introduce EscherNet, a diffusion model that can generate a flexible number of consistent target views (highlighted in **blue**) with arbitrary camera poses, based on a flexible number of reference views (highlighted in **purple**). EscherNet demonstrates precise camera control and robust generalisation across synthetic and real-world images featuring multiple objects and rich textures.

In the previous Chapter 3, we investigated efficient 3D scene representation by training vectorised neural fields at the object level. While effective, these representations are inherently object- or scene-specific and rely on dense observations for accurate 3D reconstruction. In this chapter, we shift focus towards learning generalisable 3D priors across scenes and objects, aiming to enable reconstruction or synthesis of novel scenes from only sparse observations by leveraging strong learned priors.

We introduce EscherNet, a multi-view conditioned diffusion model for view synthesis. EscherNet learns implicit and generative 3D representations coupled with a specialised camera positional encoding, allowing precise and continuous relative control of the camera transformation between an arbitrary number of reference and target views. EscherNet offers exceptional generality, flexibility, and scalability in view synthesis — it can generate more than 100 consistent target views simultaneously on a single consumer-grade GPU, despite being trained with a fixed number of 3 reference views to 3 target views. As a result, EscherNet not only addresses zero-shot novel view synthesis, but also naturally unifies single- and multi-image 3D reconstruction, combining these diverse tasks into

a single, cohesive framework. Our extensive experiments demonstrate that EscherNet achieves state-of-the-art performance in multiple benchmarks, even when compared to methods specifically tailored for each individual problem. This notable versatility opens up new directions for designing scalable neural architectures for 3D vision. Project page: <https://kxhit.github.io/EscherNet>.

4.1 Introduction

View synthesis stands as a fundamental task in computer vision and computer graphics. By allowing the re-rendering of a scene from arbitrary viewpoints based on a set of reference viewpoints, this mimics the adaptability observed in human vision. This ability is not only crucial for practical everyday tasks like object manipulation and navigation, but also plays a pivotal role in fostering human creativity, enabling us to envision and craft objects with depth, perspective, and a sense of immersion.

In this chapter, we revisit the problem of view synthesis and ask: *How can we learn a general 3D representation to facilitate scalable view synthesis?* We attempt to investigate this question from the following two observations:

i) Up until now, recent advances in view synthesis have predominantly focused on training speed and/or rendering efficiency [MESK22, KKLD23, SSC22, GKJ⁺21]. Notably, these advancements all share a common reliance on volumetric rendering for scene optimisation. Thus, all these view synthesis methods are inherently *scene-specific*, coupled with global 3D spatial coordinates. In contrast, we advocate for a paradigm shift where a 3D representation relies solely on scene colours and geometries, learning implicit representations without the need for ground-truth 3D geometry, while also maintaining independence from any specific coordinate system. This distinction is crucial for achieving scalability to overcome the constraints imposed by scene-specific encoding.

ii) View synthesis, by nature, is more suitable to be cast as a *conditional generative modelling problem*, similar to generative image in-painting [YLY⁺18, LDR⁺22]. When given only a sparse set of reference views, a desired model should provide multiple plausible predictions, leveraging the inherent stochasticity within the generative formulation and drawing insights from natural image statistics and semantic priors learned from other images and objects. As the available information

increases, the generated scene becomes more constrained, gradually converging closer to the ground-truth representation. Notably, existing 3D generative models currently only support a single reference view [LWVH⁺23, SCZ⁺23, LLZ⁺24, LGL⁺24, LXJ⁺23]. We argue that a more desirable generative formulation should flexibly accommodate varying levels of input information.

Building upon these insights, we introduce EscherNet, an image-to-image conditional diffusion model for view synthesis. EscherNet leverages a transformer architecture [VSP⁺17], employing dot-product self-attention to capture the intricate relation between both reference-to-target and target-to-target views consistencies. A key innovation within EscherNet is the design of camera positional encoding (CaPE), dedicated to representing both 4 DoF (object-centric) and 6 DoF camera poses. This encoding incorporates spatial structures into the tokens, enabling the model to compute self-attention between query and key solely based on their relative camera transformation. In summary, EscherNet exhibits these notable characteristics:

- **Consistency:** EscherNet inherently integrates view consistency thanks to the design of camera positional encoding, encouraging both *reference-to-target* and *target-to-target* view consistencies.
- **Scalability:** Unlike many existing neural rendering methods that are constrained by scene-specific optimisation, EscherNet decouples itself from any specific coordinate system and the need for ground-truth 3D geometry, without any expensive 3D operations (*e.g.* 3D convolutions or volumetric rendering), making it easier to *scale with everyday posed 2D image data*.
- **Generalisation:** Despite being trained on only a fixed number of 3 reference to 3 target views, EscherNet exhibits the capability to generate *any number of target views, with any camera poses, based on any number of reference views*. Notably, EscherNet exhibits improved generation quality with an increased number of reference views, aligning seamlessly with our original design goal.

We conduct a comprehensive evaluation across both novel view synthesis and single/multi-image 3D reconstruction benchmarks. Our findings demonstrate that EscherNet not only outperforms all 3D diffusion models in terms of generation quality but also can generate plausible view synthesis

given very limited views. This stands in contrast to these scene-specific neural rendering methods such as InstantNGP [MESK22] and Gaussian Splatting [KKLD23], which often struggle to generate meaningful content under such constraints. This underscores the effectiveness of our method’s simple yet scalable design, offering a promising avenue for advancing view synthesis and 3D vision as a whole.

4.2 Related Work

Neural 3D Representations Early works in neural 3D representation learning focused on directly optimising on 3D data, using representations such as voxels [MS15] and point clouds [QSMG17, QYSG17], for explicit 3D representation learning. Alternatively, another line of works focused on training neural networks to map 3D spatial coordinates to signed distance functions [PFS⁺19] or occupancies [MON⁺19, PNM⁺20], for implicit 3D representation learning. However, all these methods heavily rely on ground-truth 3D geometry, limiting their applicability to small-scale synthetic 3D data [WSK⁺15, CFG⁺15].

To accommodate a broader range of data sources, differentiable rendering functions [NMOG20, SZW19] have been introduced to optimise neural implicit shape representations with multi-view posed images. More recently, NeRF [MST⁺20a] paved the way to a significant enhancement in rendering quality compared to these methods by optimising MLPs to encode 5D radiance fields. In contrast to tightly coupling 3D scenes with spatial coordinates, we introduce EscherNet as an alternative for 3D representation learning by optimising a neural network to learn the interaction between multi-view posed images, independent of any coordinate system.

Novel View Synthesis The success of NeRF has sparked a wave of follow-up methods that address faster training and/or rendering efficiency, by incorporating different variants of space discretisation [GKJ⁺21, HSM⁺21, CXG⁺22], codebooks [TET⁺22], and encodings using hash tables [MESK22] or Gaussians [KKLD23].

To enhance NeRF’s generalisation ability across diverse scenes and in a few-shot setting, Pixel-

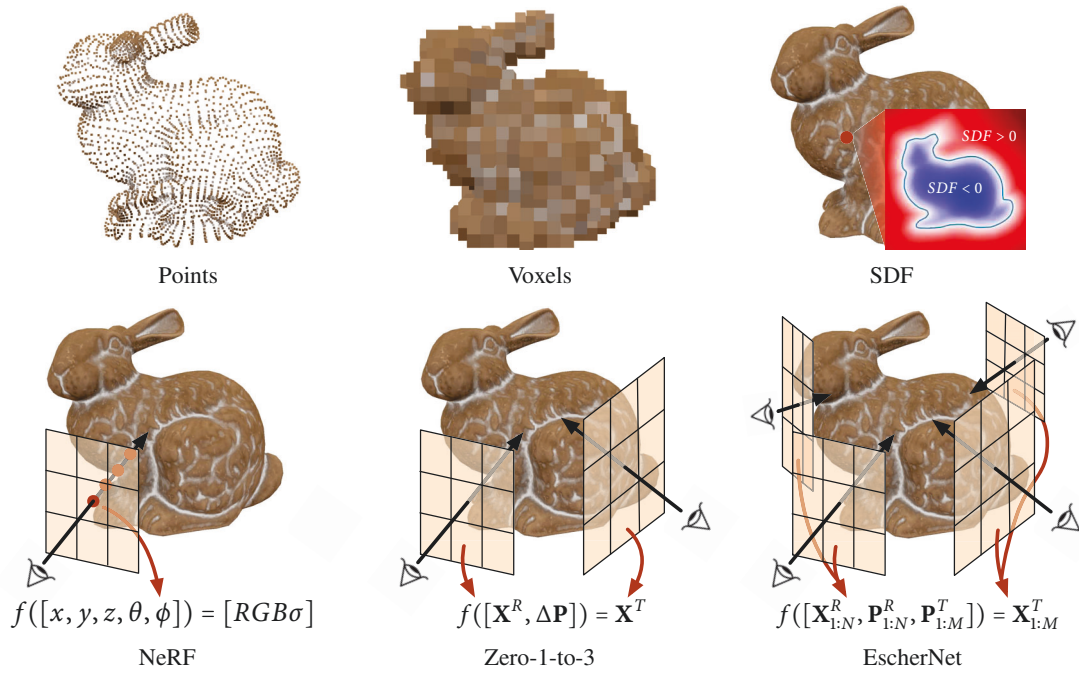


Figure 4.2: **3D Representations Overview.** EscherNet generates a set of M target views $\mathbf{X}_{1:M}^T$ based on their camera poses $\mathbf{P}_{1:M}^T$, leveraging information gained from a set of N reference views $\mathbf{X}_{1:N}^R$ and their camera poses $\mathbf{P}_{1:N}^R$. EscherNet presents a new way of learning implicit 3D representations by only considering the relative camera transformation between the camera poses of $\mathbf{P}^R$ and $\mathbf{P}^T$, making it easier to scale with multi-view posed images, independent of any specific coordinate systems.

NeRF [YYTK21] attempts to learn a scene prior by jointly optimising multiple scenes, but it is constrained by the high computational demands required by volumetric rendering. Various other approaches have addressed this issue by introducing regularisation techniques, such as incorporating low-level priors from local patches [NBM⁺22], ensuring semantic consistency [JTA21], considering adjacent ray frequency [YPW23], and incorporating depth signals [DLZR22]. In contrast, EscherNet encodes scenes directly through the image space, enabling the learning of more generalised scene priors through large-scale datasets.

3D Diffusion Models The emergence of 2D generative diffusion models has shown impressive capabilities in generating realistic objects and scenes [HJA20, RBL⁺22]. This progress has inspired the early design of text-to-3D diffusion models, such as DreamFusion [PJBM23] and Magic3D [LGT⁺23], by optimising a radiance field guided by score distillation sampling (SDS) from these pre-trained 2D diffusion models. However, SDS necessitates computationally intensive iterative optimisation, often requiring up to an hour for convergence. Additionally, these methods, including recently proposed

image-to-3D generation approaches [DJQ⁺23, XJW⁺23, MKLRV23], frequently yield unrealistic 3D generation results due to their limited 3D understanding, giving rise to challenges such as the multi-face Janus problem.

To integrate 3D priors more efficiently, an alternative approach involves training 3D generative models directly on 3D datasets, employing representations such as point clouds [NJD⁺22] or neural fields [EMS⁺23, CGC⁺23, JN23]. However, this design depends on 3D operations, such as 3D convolution and volumetric rendering, which are computationally expensive and challenging to scale.

To address this issue, diffusion models trained on multi-view posed data have emerged as a promising direction, designed with no 3D operations. Zero-1-to-3 [LWVH⁺23] stands out as a pioneering work, learning view synthesis from paired 2D posed images rendered from large-scale 3D object datasets [DSS⁺23, DLW⁺23]. However, its capability is limited to generating a single target view conditioned on a single reference view. Recent advancements in multi-view diffusion models [SWY⁺24, YWL⁺24, SCZ⁺23, LLZ⁺24, LGL⁺24, LXJ⁺23] focused on 3D generation and can only generate a fixed number of target views with fixed camera poses. In contrast, EscherNet can generate an unrestricted number of target views with arbitrary camera poses, offering superior flexibility in view synthesis.

4.3 EscherNet

Problem Formulation and Notation In EscherNet, we recast the view synthesis as a conditional generative modelling problem, formulated as:

$$\mathcal{X}^T \sim p(\mathcal{X}^T | \mathcal{X}^R, \mathcal{P}^R, \mathcal{P}^T). \quad (4.1)$$

Here, $\mathcal{X}^T = \{\mathbf{X}_{1:M}^T\}$ and $\mathcal{P}^T = \{\mathbf{P}_{1:M}^T\}$ represent a set of M target views $\mathbf{X}_{1:M}^T$ with their global camera poses $\mathbf{P}_{1:M}^T$. Similarly, $\mathcal{X}^R = \{\mathbf{X}_{1:N}^R\}$ and $\mathcal{P}^R = \{\mathbf{P}_{1:N}^R\}$ represent a set of N reference views $\mathbf{X}_{1:N}^R$ with their global camera poses $\mathbf{P}_{1:N}^R$. Both N and M can take on arbitrary values during both model training and inference.

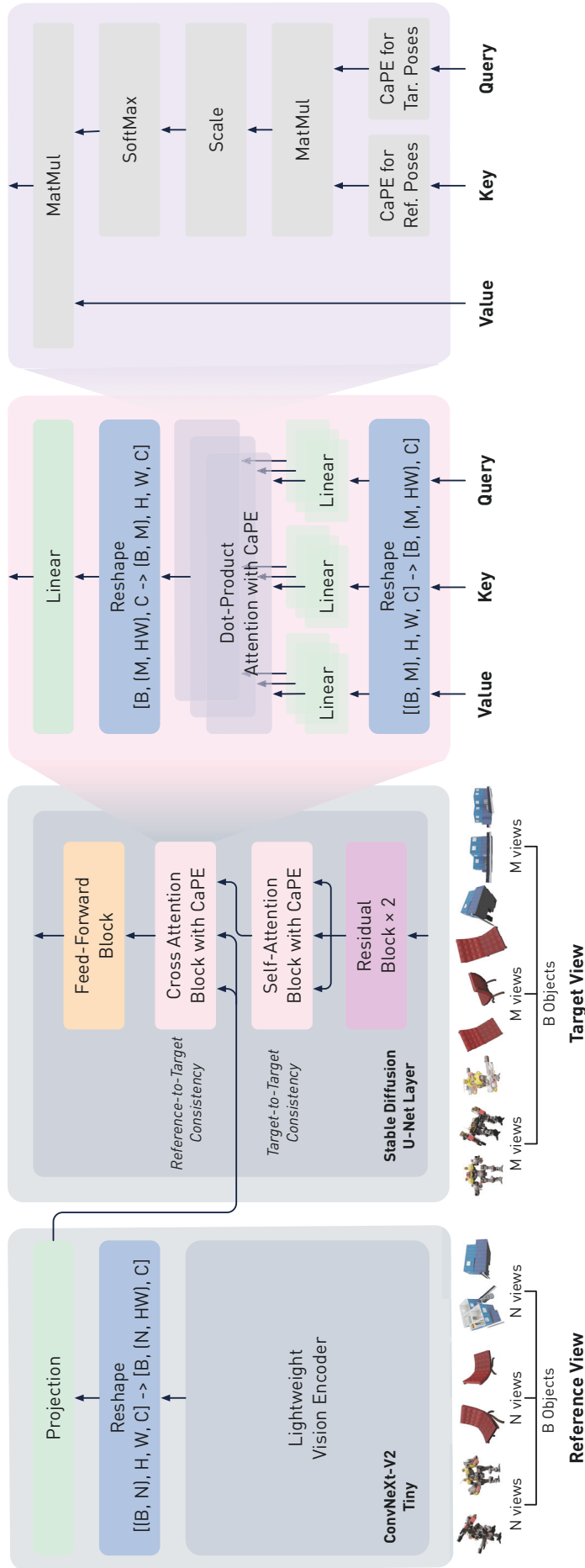


Figure 4.3: **EscherNet Architecture.** EscherNet adopts the Stable Diffusion architectural design with minimal but important modifications. The lightweight vision encoder captures both high-level and low-level signals from N reference views. In U-Net, we apply self-attention within M target views to encourage target-to-target consistency, and cross-attention within M target and N reference views (encoded by the image encoder) to encourage reference-to-target consistency. In each attention block, CaPE is employed for the key and query, allowing the attention map to learn with relative camera poses, independent of specific coordinate systems.

We propose a neural architecture design, such that the generation of each target view $\mathbf{X}_i^T \in \mathcal{X}^T$ solely depends on its relative camera transformation to the reference views $(\mathbf{P}_j^R)^{-1}\mathbf{P}_i^T, \forall \mathbf{P}_j^R \in \mathcal{P}^R$, introduced next.

4.3.1 Architecture Design

We design EscherNet following two key principles: i) It builds upon an existing 2D diffusion model, inheriting its strong web-scale prior through large-scale training, and ii) It encodes camera poses for each view/image, similar to how language models encode token positions for each token. So our model can naturally handle an arbitrary number of views for *any-to-any view synthesis*. The architecture is shown in Fig. 4.3.

Multi-View Generation EscherNet can be seamlessly integrated with any 2D diffusion model with a transformer architecture, with *no additional learnable parameters*. In this work, we design EscherNet by adopting a latent diffusion architecture, specifically `StableDiffusion v1.5` [RBL⁺22]. This choice enables straightforward comparisons with numerous 3D diffusion models that also leverage the same backbone (more details in the experiment section).

To tailor the Stable Diffusion model, originally designed for text-to-image generation, to multi-view generation as applied in EscherNet, several key modifications are implemented. In the original Stable Diffusion’s denoiser U-Net, the self-attention block was employed to learn interactions within different patches within the same image. In EscherNet, we re-purpose this self-attention block to facilitate learning interactions within distinct patches across M different target views, thereby ensuring target-to-target consistency. Likewise, the cross-attention block, originally used to integrate textual information into image patches, is repurposed in EscherNet to learn interactions within N reference to M target views, ensuring reference-to-target consistency.

Conditioning Reference Views In view synthesis, it is crucial that the conditioning signals accurately capture both the high-level semantics and low-level texture details present in the reference views. Previous works in 3D diffusion models [LWVH⁺23, LLZ⁺24] have employed the strategy

of encoding high-level signals through a frozen CLIP pre-trained ViT [RKH⁺21] and encoding low-level signals by concatenating the reference image into the input of the U-Net of Stable Diffusion. However, this design choice inherently constrains the model to handle only one single view.

In EscherNet, we choose to incorporate both high-level and low-level signals in the conditioning image encoder, representing reference views as sets of tokens. This design choice allows our model to maintain flexibility in handling a variable number of reference views. Early experiments have confirmed that using a frozen CLIP-ViT alone may fail to capture low-level textures, preventing the model from accurately reproducing the original reference views given the same reference view poses as target poses. While fine-tuning the CLIP-ViT could address this issue, it poses challenges in terms of training efficiency. Instead, we opt to fine-tune a lightweight vision encoder, specifically ConvNeXt_{v2}-Tiny [WDH⁺23], which is a highly efficient CNN architecture. This architecture is employed to compress our reference views to smaller resolution image features. We treat these image features as conditioning tokens, effectively representing each reference view. This configuration has proven to be sufficient in our experiments, delivering superior results in generation quality while simultaneously maintaining high training efficiency.

4.3.2 Camera Positional Encoding (CaPE)

To encode camera poses efficiently and accurately into reference and target view tokens within a transformer architecture, we introduce Camera Positional Encoding (CaPE), drawing inspiration from recent advancements in the language domain. We first briefly examine the distinctions between these two domains.

- In language, token positions (associated with each word) follow a *linear and discrete* structure, and their length can be *infinite*. Language models are typically trained with fixed maximum token counts (known as context length), and it remains an ongoing research challenge to construct a positional encoding that enables the model to behave reasonably beyond this fixed context length [PQFS24, HWP⁺24].
- In 3D vision, token positions (associated with each camera) follow a *cyclic, continuous, and*

bounded structure for rotations and a *linear, continuous, and unbounded* structure for translations. Importantly, unlike the language domain where the token position always starts from zero, there are no *standardised absolute global camera poses* in a 3D space. The relationship between two views depends solely on their relative camera transformation.

We now present two distinct designs for spatial position encoding, representing camera poses using 4 DoF for object-centric rendering and 6 DoF for the generic case, respectively. Our design strategy involves directly applying a transformation on global camera poses embedded in the token feature, which allows the dot-product attention to directly encode the relative camera transformation, independent of any coordinate system.

4 DoF CaPE In the case of 4 DoF camera poses, we adopt a spherical coordinate system, similar to [LWVH⁺23, LLZ⁺24], denoted as $\mathbf{P} = \{\alpha, \beta, \gamma, r\}$ including azimuth, elevation, camera orientation along the look-at direction, and camera distance (radius), each position component is *disentangled*.

Mathematically, the position encoding function $\pi(\mathbf{v}, \mathbf{P})$, characterised by its d -dimensional token feature $\mathbf{v} \in \mathbb{R}^d$ and pose $\mathbf{P}$, should satisfy the following conditions:

$$\langle \pi(\mathbf{v}_1, \theta_1), \pi(\mathbf{v}_2, \theta_2) \rangle = \langle \pi(\mathbf{v}_1, \theta_1 - \theta_2), \pi(\mathbf{v}_2, 0) \rangle, \quad (4.2)$$

$$\langle \pi(\mathbf{v}_1, r_1), \pi(\mathbf{v}_2, r_2) \rangle = \langle \pi(\mathbf{v}_1, r_1/r_2), \pi(\mathbf{v}_2, 1) \rangle. \quad (4.3)$$

Here $\langle \cdot, \cdot \rangle$ represents the dot product operation, $\theta_{1,2} \in \{\alpha, \beta, \gamma\}$, within $\alpha, \gamma \in [0, 2\pi), \beta \in [0, \pi)$, and $r_{1,2} > 0$. Essentially, the relative 4 DoF camera transformation is decomposed to the relative angle difference in rotation and the relative scale difference in view radius.

Notably, Eq. 4.2 aligns with the formula of rotary position encoding (RoPE) [SLP⁺21] derived in the language domain. Given that $\log(r_1) - \log(r_2) = \log(s \cdot r_1) - \log(s \cdot r_2)$ (for any scalar $s > 0$), we may elegantly combine both Eq. 4.2 and Eq. 4.3 in a unified formulation using the design strategy in RoPE by transforming feature vector $\mathbf{v}$ with a block diagonal rotation matrix $\phi(\mathbf{P})$ encoding $\mathbf{P}$.

– **4 DoF CaPE**: $\pi(\mathbf{v}, \mathbf{P}) = \phi(\mathbf{P})\mathbf{v}$,

$$\phi(\mathbf{P}) = \begin{bmatrix} \Psi & 0 & \cdots & 0 \\ 0 & \Psi & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \Psi \end{bmatrix}, \quad \Psi = \begin{bmatrix} \Psi_\alpha & 0 & \cdots & 0 \\ 0 & \Psi_\beta & 0 & \vdots \\ \vdots & 0 & \Psi_\gamma & 0 \\ 0 & \cdots & 0 & \Psi_r \end{bmatrix}. \quad (4.4)$$

$$\text{Rotation: } \Psi_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \quad (4.5)$$

$$\text{View Radius: } \Psi_r = \begin{bmatrix} \cos(f(r)) & -\sin(f(r)) \\ \sin(f(r)) & \cos(f(r)) \end{bmatrix}, \quad (4.6)$$

$$\text{where } f(r) = \pi \frac{\log r - \log r_{\min}}{\log r_{\max} - \log r_{\min}} \in [0, \pi]. \quad (4.7)$$

Here, $\dim(\mathbf{v}) = d$ should be divisible by $2|\mathbf{P}| = 8$. Note, it's crucial to apply Eq. 4.7 to constrain $\log r$ within the range of rotation $[0, \pi]$, so we ensure the dot product monotonically corresponds to its scale difference.

6 DoF CaPE In the case of 6 DoF camera poses, denoted as $\mathbf{P} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \in SE(3)$, each position component is *entangled*, implying that we are not able to reformulate as a multi-dimensional position as in 4 DoF camera poses.

Mathematically, the position encoding function $\pi(\mathbf{v}, \mathbf{P})$ should now satisfy the following condition:

$$\langle \pi(\mathbf{v}_1, \mathbf{P}_1), \pi(\mathbf{v}_2, \mathbf{P}_2) \rangle = \langle \pi(\mathbf{v}_1, \mathbf{P}_2^{-1}\mathbf{P}_1), \pi(\mathbf{v}_2, \mathbf{I}) \rangle. \quad (4.8)$$

Let's apply a similar strategy as used in 4 DoF CaPE, which increases the dimensionality of $\mathbf{P} \in \mathbb{R}^{4 \times 4}$ to $\phi(\mathbf{P}) \in \mathbb{R}^{d \times d}$ by reconstructing it as a block diagonal matrix, with each diagonal element being $\mathbf{P}$. Since $\phi(\mathbf{P})$ also forms a real Lie group, we may construct $\pi(\cdot, \cdot)$ for a key and query using the

following equivalence:

$$(\phi(\mathbf{P}_2^{-1}\mathbf{P}_1)\mathbf{v}_1)^\top (\phi(\mathbf{I})\mathbf{v}_2) = (\mathbf{v}_1^\top \phi(\mathbf{P}_1^\top \mathbf{P}_2^{-\top})) \mathbf{v}_2 \quad (4.9)$$

$$= (\mathbf{v}_1^\top \phi(\mathbf{P}_1^\top))(\phi(\mathbf{P}_2^{-\top})\mathbf{v}_2) = (\phi(\mathbf{P}_1)\mathbf{v}_1)^\top (\phi(\mathbf{P}_2^{-\top})\mathbf{v}_2) \quad (4.10)$$

$$= \langle \pi(\mathbf{v}_1, \phi(\mathbf{P}_1)), \pi(\mathbf{v}_2, \phi(\mathbf{P}_2^{-\top})) \rangle. \quad (4.11)$$

– **6 DoF CaPE**: $\pi(\mathbf{v}, \mathbf{P}) = \phi(\mathbf{P})\mathbf{v}$,

$$\phi(\mathbf{P}) = \begin{bmatrix} \Psi & 0 & \cdots & 0 \\ 0 & \Psi & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \Psi \end{bmatrix}, \quad \Psi = \begin{cases} \mathbf{P} & \text{if key} \\ \mathbf{P}^{-\top} & \text{if query} \end{cases}. \quad (4.12)$$

Here, $\dim(\mathbf{v}) = d$ should be divisible by $\dim(\mathbf{P}) = 4$. Similarly, we need to re-scale the translation $\mathbf{t}$ for each scene within a unit range for efficient model training.

In both 4 and 6 DoF CaPE implementation, we can efficiently perform matrix multiplication by simply reshaping the vector $\mathbf{v}$ to match the dimensions of Ψ (8 for 4 DoF, 4 for 6 DoF), ensuring faster computation.

Python Implementation of CaPE We provide Python implementations of CaPE for both 4DoF (Listing 4.3.2) and 6DoF (Listing 4.3.2). Similar to RoPE [SLP⁺21], CaPE can be seamlessly integrated into the Transformer’s attention mechanism without introducing additional parameters or computational overhead.

```

def compute_4dof_cape(v, P, s):
    """
    :param v: input feature vector with its dimension must be divisible by 8
    :param P: list = [alpha, beta, gamma, r]
    :param s: a small scalar for radius
    :return: rotated v with its corresponding camera pose P
    """
    v = v.reshape([-1, 8])
    psi = np.zeros([8, 8])
    for i in range(4):
        if i < 3:
            psi[2 * i:2 * (i + 1), 2 * i:2 * (i + 1)] = \
                np.array([[np.cos(P[i]), -np.sin(P[i])], [np.sin(P[i]), np.cos(P[i])]])
        else:
            psi[2 * i:2 * (i + 1), 2 * i:2 * (i + 1)] = \
                np.array([[np.cos(s * np.log(P[i])), -np.sin(s * np.log(P[i]))],
                           [np.sin(s * np.log(P[i])), np.cos(s * np.log(P[i]))]])

    return v.dot(psi).reshape(-1)

```

Listing 4.1: Python implementation for 4 DoF CaPE.

```

def compute_6dof_cape(v, P, s=0.001, key=True):
    """
    :param v: input feature vector with its dimension must be divisible by 4
    :param P: 4 x 4 SE3 matrix
    :param s: a small scalar for translation
    :return: rotated v with its corresponding camera pose P
    """
    v = v.reshape([-1, 4])
    P[:3, 3] *= s
    psi = P if key else np.linalg.inv(P).T
    return v.dot(psi).reshape(-1)

```

Listing 4.2: Python implementation for 6 DoF CaPE.

4.4 Experiments

4.4.1 Implementation Details

Training Datasets In this work, we focus on object-centric view synthesis, training our model on Objaverse-1.0 which consists of 800K objects [DSS⁺23]. This setting allows us to fairly compare with all other 3D diffusion model baselines trained on the same dataset. We adopt the same training data used in Zero-1-to-3 [LWVH⁺23], which contains 12 randomly rendered views per object with randomised environment lighting. To ensure the data quality, we filter out empty rendered images, which make up roughly 1% of the training data.

We trained and reported results using EscherNet with both 4 DoF and 6 DoF CaPE. Our observations revealed that 6 DoF CaPE exhibits a slightly improved performance, which we attribute to its more compressed representation space. However, empirically, we found that 4 DoF CaPE yields visually more consistent results when applied to real-world images. Considering that the training data is confined within a 4 DoF object-centric setting, we present EscherNet with 4 DoF CaPE in the main experiments and provide the comparison with 6DoF later.

Optimisation and Implementation EscherNet is trained using the AdamW optimiser [LH19] with a learning rate of $1 \cdot 10^{-4}$ and weight decay of 0.01 for $[256 \times 256]$ resolution images. We incorporate cosine annealing, reducing the learning rate to $1 \cdot 10^{-5}$ over a total of 100,000 training steps, while linearly warming up for the initial 1000 steps. To speed up training, we implement automatic mixed precision with a precision of `bfloat16` and employ gradient checkpointing. Our training batches consist of 3 reference views and 3 target views randomly sampled with replacement from 12 views for each object, with a total batch size of 672 (112 batches per GPU). The entire model training process takes 1 week on 6 NVIDIA A100 80GB GPUs.

4.4.2 Evaluation Details

In all experiments, we re-evaluate the baseline models by using their officially open-sourced checkpoints on the same set of reference views for a fair comparison.

Metrics To evaluate view synthesis quality in 2D metrics, we employ PSNR, SSIM [WBSS04], and LPIPS [ZIE⁺18]. For 3D metrics used in 3D generation, we employ Chamfer Distance and Volume IoU. To ensure a fair and efficient evaluation process, each baseline method and our approach are executed only once per scene per viewpoint. This practice has proven to provide stable averaged results across multiple scenes and viewpoints.

In NeRF Synthetic Dataset [MST⁺20a], we consider and evaluate all 8 scenes provided in the original dataset. To assess performance with varying numbers of reference views, we train all baseline methods and our approach using the same set of views randomly sampled from the training set. The evaluation is conducted on all target views defined in the test sets across all 8 scenes (with 200 views per scene). For InstantNGP [MESK22], we run 10k steps ($\approx$ 1min) for each scene. For 3D Gaussian Splatting [KKLD23], we run 5k steps ($\approx$ 2min) for each scene.

In Google Scanned Dataset (GSO) [DFK⁺22], we evaluate the same 30 objects chosen by SyncDreamer [LLZ⁺24]. For each object, we render 25 views with randomly generated camera poses and a randomly generated environment lighting condition to construct our test set. For each object, we choose the first 10 images as our reference views and the subsequent 15 images as our target views for evaluation. It’s crucial to note that all reference and target views are rendered with random camera poses, establishing a more realistic and challenging evaluation setting compared to the evaluation setups employed in other baselines: *e.g.* SyncDreamer uses an evenly distributed environment lighting to render all GSO data, and the reference view for each object is manually selected based on human preference.¹ Additionally, the evaluated target view is also manually selected based on human preference chosen among four independent generations.²

¹<https://github.com/liuyuan-pal/SyncDreamer/issues/21>

²<https://github.com/liuyuan-pal/SyncDreamer/issues/21#issuecomment-1770345260>

	Training Data	# Ref. Views	GSO-30			RTMV		
			PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
RealFusion	-	1	12.76	0.758	0.382	-	-	-
Zero123	800K	1	18.51	0.856	0.127	10.16	0.505	0.418
Zero123-XL	10M	1	18.93	0.856	0.124	10.59	0.520	0.401
EscherNet	800k	1	20.24	0.884	0.095	10.56	0.518	0.410
EscherNet	800k	2	22.91	0.908	0.064	12.66	0.585	0.301
EscherNet	800k	3	24.09	0.918	0.052	13.59	0.611	0.258
EscherNet	800k	5	25.09	0.927	0.043	14.52	0.633	0.222
EscherNet	800k	10	25.90	0.935	0.036	15.55	0.657	0.185

Table 4.1: **Novel View Synthesis Results on GSO and RTMV Datasets.** EscherNet outperforms Zero-1-to-3-XL with significantly less training data and RealFusion without extra SDS optimisation. Additionally, EscherNet’s performance exhibits further improvement with the inclusion of more reference views.

In evaluating 3D generation, we randomly sample 4096 points evenly distributed from the generated 3D mesh or point cloud across all methods. Each method’s generated mesh is aligned to the ground-truth mesh using the camera pose of the reference views. Specifically in Point-E [NJD⁺22] and Shape-E [JN23], we rotate 90/180 degrees along each x/y/z axis to determine the optimal alignment for the final mesh pose. Our evaluation approach again differs from SyncDreamer, which initially projects the 3D mesh into their fixed 16 generated views to obtain depth maps. Then, points are sampled from these depth maps for the final evaluation.³

In RTMV Dataset [TME⁺22], we follow the evaluation setting used in Zero-1-to-3 [LWVH⁺23], which consists of 10 complex scenes featuring a pile of multiple objects from the GSO dataset. Similar to the construction of our GSO test set, we then randomly select a fixed subset of the first 10 images as our reference views and the subsequent 10 views as our target views for evaluation.

4.4.3 Results on Novel View Synthesis

We evaluate EscherNet in the novel view synthesis task on the Google Scanned Objects dataset (GSO) [DFK⁺22] and the RTMV dataset [TME⁺22], comparing with 3D diffusion models for view

³<https://github.com/liuyuan-pal/SyncDreamer/issues/44>

	# Reference Views (Less → More)							
	1	2	3	5	10	20	50	100
InstantNGP (Scene Specific Training)								
PSNR↑	10.92	12.42	14.27	18.17	22.96	24.99	26.86	27.30
SSIM↑	0.449	0.521	0.618	0.761	0.881	0.917	0.946	0.953
LPIPS↓	0.627	0.499	0.391	0.228	0.091	0.058	0.034	0.031
GaussianSplatting (Scene Specific Training)								
PSNR↑	9.44	10.78	12.87	17.09	23.04	25.34	26.98	27.11
SSIM↑	0.391	0.432	0.546	0.732	0.876	0.919	0.942	0.944
LPIPS↓	0.610	0.541	0.441	0.243	0.085	0.054	0.041	0.041
EscherNet (Zero Shot Inference)								
PSNR↑	13.36	14.95	16.19	17.16	17.74	17.91	18.05	18.15
SSIM↑	0.659	0.700	0.729	0.748	0.761	0.765	0.769	0.771
LPIPS↓	0.291	0.208	0.161	0.127	0.114	0.106	0.099	0.097

Table 4.2: **Novel View Synthesis Results on NeRF Synthetic Dataset.** EscherNet outperforms both InstantNGP and Gaussian Splatting when provided with fewer than five reference views while requiring no scene-specific optimisation. However, as the number of reference views increases, both methods show a more significant improvement in rendering quality.

synthesis, such as Zero-1-to-3 [LWVH⁺23] and RealFusion [MKLRV23] (primarily for generation quality with minimal reference views). Also, we evaluate on the NeRF Synthetic dataset [MST⁺20a], comparing with SoTA scene-specific neural rendering methods, such as InstantNGP [MESK22] and 3D Gaussian Splatting [KKLD23] (primarily for rendering accuracy with multiple reference views).

Notably, many other 3D diffusion models [LLZ⁺24, LGL⁺24, SCZ⁺23, YWL⁺24, LXJ⁺23] prioritise 3D generation rather than view synthesis. This limitation confines them to predicting target views with *fixed target poses*, making them not directly comparable.

Compared to 3D Diffusion Models In Tab. 4.1 and Fig. 4.4, we show that EscherNet significantly outperforms 3D diffusion baselines, by a large margin, both quantitatively and qualitatively. Particularly, we outperform Zero-1-to-3-XL despite it being trained on $\times 10$ more training data, and RealFusion despite it requiring expensive score distillation for iterative scene optimisation [PJB23]. It’s worth highlighting that Zero-1-to-3 by design is inherently limited to generating a single target view and cannot ensure self-consistency across multiple target views, while EscherNet can generate multiple consistent target views jointly and provides more precise camera control.

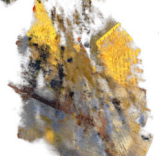
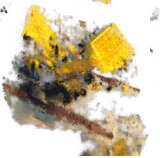
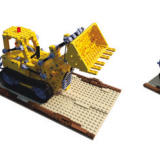
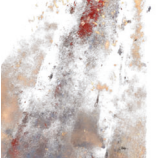
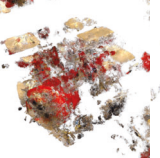
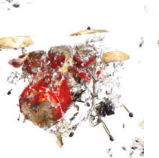
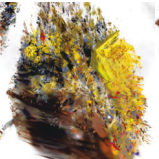
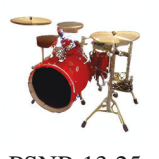
# Reference Views (Less → More)							
1	2	3	5	10	20	50	100
InstantNGP (Scene Specific Training)							
							
PSNR 9.45	PSNR 11.41	PSNR 13.64	PSNR 19.30	PSNR 23.14	PSNR 26.18	PSNR 28.54	PSNR 28.87
							
PSNR 10.37	PSNR 11.72	PSNR 12.82	PSNR 15.58	PSNR 19.71	PSNR 21.28	PSNR 23.09	PSNR 23.78
3D Gaussian Splatting (Scene Specific Training)							
							
PSNR 8.07	PSNR 9.16	PSNR 11.72	PSNR 17.32	PSNR 24.19	PSNR 25.34	PSNR 26.98	PSNR 29.01
							
PSNR 9.14	PSNR 10.63	PSNR 11.43	PSNR 14.81	PSNR 20.15	PSNR 22.88	PSNR 23.49	PSNR 23.51
EscherNet (Zero Shot Inference)							
							
PSNR 10.86	PSNR 10.80	PSNR 15.51	PSNR 17.07	PSNR 17.40	PSNR 17.38	PSNR 17.77	PSNR 17.85
							
PSNR 10.10	PSNR 13.25	PSNR 13.43	PSNR 14.33	PSNR 14.97	PSNR 15.65	PSNR 15.70	PSNR 15.90

Table 4.3: Novel View Synthesis on NeRF Synthetic Dataset. We report the average PSNR per scene, conditioned on the respective number of reference views. EscherNet generates plausible view synthesis even when provided with very limited reference views, while neural rendering methods fail to generate any meaningful content. However, when we have more than 10 reference views, scene-specific methods exhibit a substantial improvement in rendering quality.

Compared to Neural Rendering Methods In Tab. 4.2 and Tab. 4.3, we show that EscherNet again offers plausible view synthesis in a zero-shot manner, without scene-specific optimisation required

	# Ref. Views	Chamfer Dist. ↓	Volume IoU ↑
Point-E	1	0.0447	0.2503
Shape-E	1	0.0448	0.3762
One2345	1	0.0632	0.4209
One2345-XL	1	0.0667	0.4016
DreamGaussian	1	0.0605	0.3757
DreamGaussian-XL	1	0.0459	0.4531
SyncDreamer	1	0.0400	0.5220
NeuS	3	0.0366	0.5352
NeuS	5	0.0245	0.6742
NeuS	10	0.0195	0.7264
EscherNet	1	0.0314	0.5974
EscherNet	2	0.0215	0.6868
EscherNet	3	0.0190	0.7189
EscherNet	5	0.0175	0.7423
EscherNet	10	0.0167	0.7478

Table 4.4: **3D Reconstruction Results on GSO.** EscherNet outperforms all other image-to-3D baselines in generating more visually appealing and accurate 3D geometry, particularly when conditioned on multiple reference views.

by both InstantNGP and 3D Gaussian Splatting. Notably, EscherNet leverages a generalised understanding of objects acquired through large-scale training, allowing it to interpret given views both semantically and spatially, even when conditioned on a limited number of reference views. However, with an increase in the number of reference views, both InstantNGP and 3D Gaussian Splatting exhibit a significant improvement in the rendering quality. To achieve a photo-realistic neural rendering while retaining the advantages of a generative formulation remains an important research challenge.

4.4.4 Results on 3D Generation

In this section, we perform single/few-image 3D generation on the GSO dataset. We compare with SoTA 3D generation baselines: Point-E [NJD⁺22] for direct point cloud generation, Shape-E [JN23] for direct NeRF generation, DreamGaussian [JN23] for optimising 3D Gaussian [KKLD23] with SDS guidance, One-2-3-45 [LXJ⁺23] for decoding an SDF using multiple views predicted from Zero-1-to-3, and SyncDreamer [LLZ⁺24] for fitting an SDF using NeuS [WLL⁺21] from 16 consistent fixed generated views. We additionally include NeuS trained on reference views for few-image 3D

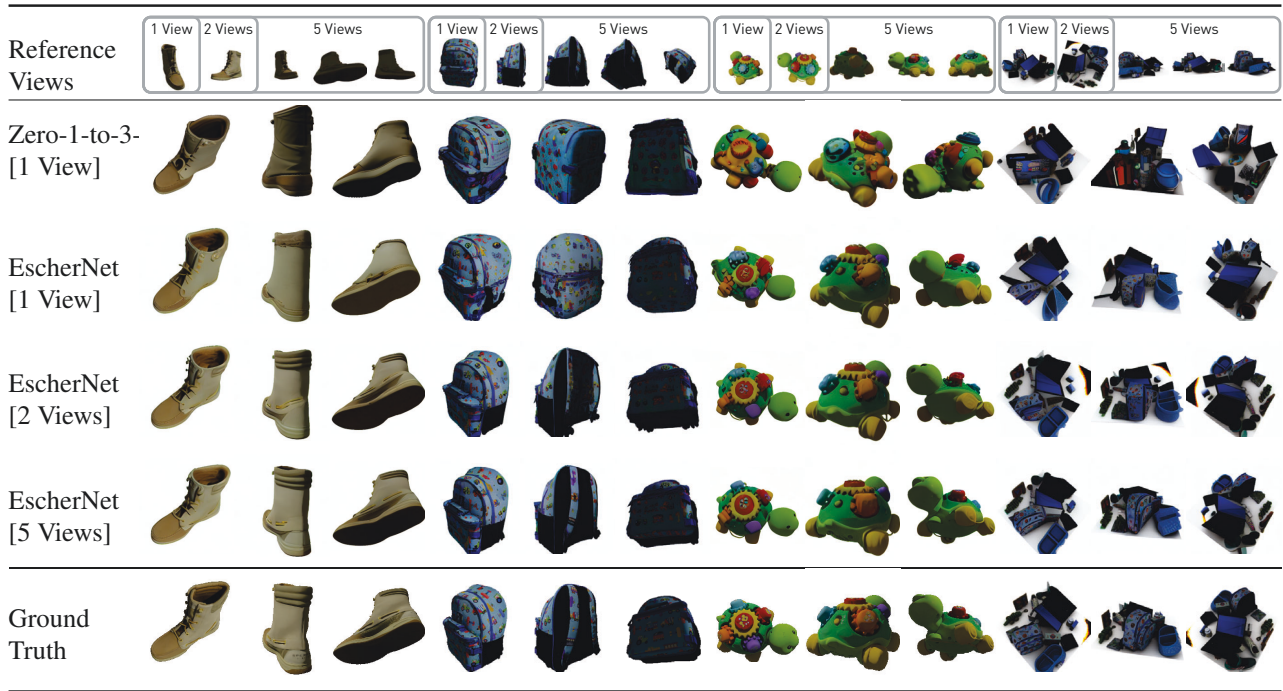


Figure 4.4: **Novel View Synthesis Visualisation on GSO and RTMV Datasets.** EscherNet outperforms Zero-1-to-3-XL, delivering superior generation quality and finer camera control. Notably, when conditioned with additional views, EscherNet exhibits an enhanced resemblance of the generated views to ground-truth textures, revealing more refined texture details such as in the backpack straps and turtle shell.

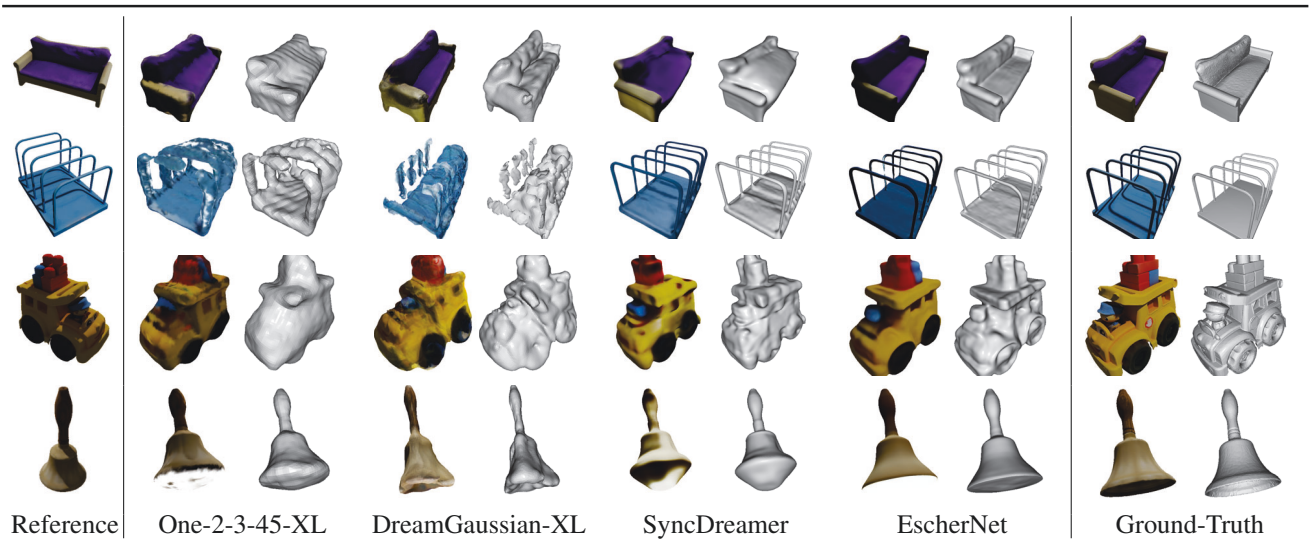


Figure 4.5: **Single View 3D Reconstruction Visualisation on GSO.** EscherNet’s ability to generate dense and consistent novel views significantly improves the reconstruction of complete and well-constrained 3D geometry. In contrast, One-2-3-45-XL and DreamGaussian-XL, despite leveraging a significantly larger pre-trained model, tend to produce over-smoothed and noisy reconstructions; SyncDreamer, constrained by sparse fixed-view synthesis, struggles to tightly constrain geometry, particularly in areas in sofa and the bottom part of the bell.

reconstruction baselines.

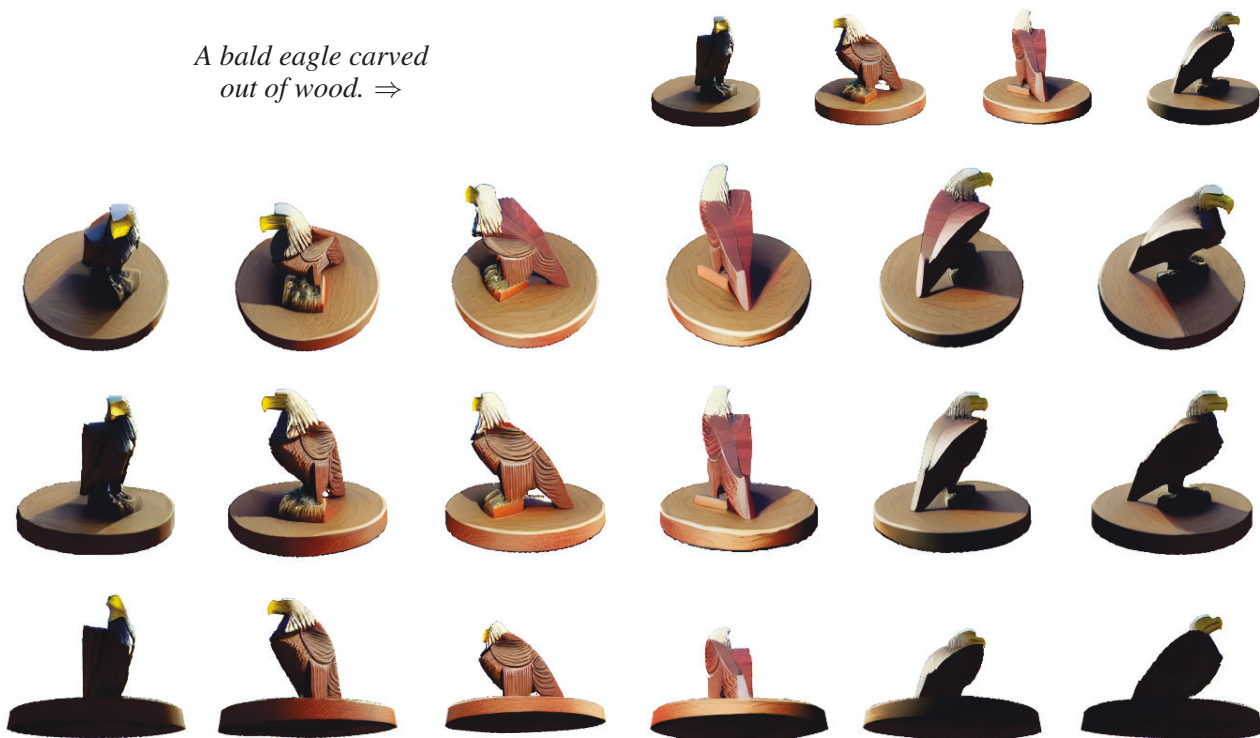
Given any reference views, EscherNet can generate multiple 3D consistent views, allowing for the straightforward adoption with NeuS [WLL⁺21] for 3D reconstruction. We generate 36 fixed views, varying the azimuth from 0° to 360° with a rendering every 30° at a set of elevations (-30°, 0°, 30°), which serve as inputs for our NeuS reconstruction.

In Tab. 4.4 and Fig. 4.5, we show that EscherNet stands out by achieving significantly superior 3D reconstruction quality compared to other image-to-3D generative models. Specifically, EscherNet demonstrates an approximate 25% improvement in Chamfer distance over SyncDreamer, considered as the current best model, when conditioned on a single reference view, and a 60% improvement when conditioned on 10 reference views. This impressive performance is attributed to EscherNet’s ability to flexibly handle any number of reference and target views, providing comprehensive and accurate constraints for 3D geometry. In contrast, SyncDreamer faces challenges due to sensitivity to elevation angles and constraints imposed by a fixed 30° elevation angle by design, thus hindering learning a holistic representation of complex objects. This limitation results in degraded reconstruction, particularly evident in the lower regions of the generated geometry.

4.4.5 Results on Text-to-3D Generation

EscherNet’s flexibility in accommodating any number of reference views enables a straightforward approach to the text-to-3D generation problem by breaking it down into two stages: text-to-image, relying on any off-the-shelf text-to-image generative model, and then image-to-3D, relying on EscherNet. In Fig. 4.6 and Tab. 4.5, we present visual results of dense novel view generation using a text-to-4view model with MVDream [SWY⁺24] and a text-to-image model with SDXL [PEL⁺24]. Notably, even when dealing with out-of-distribution and counterfactual content, EscherNet generates consistent 3D novel views with appealing textures.

A bald eagle carved out of wood. $\Rightarrow$



A robot made of vegetables, 4K. $\Rightarrow$



Figure 4.6: **Text-to-3D Visualisation with MVDream (up) and SDXL (bottom).** EscherNet offers compelling and realistic view synthesis for synthetic images generated with user-provided text prompts.

A robot made of vegetables. 	           
A nurse corgi. 	           
A cute steampunk elephant. 	           
A bull dog wearing a black pirate hat. 	           
An astronaut riding a horse. 	           
Medieval House, grass, medieval-decor, 3d asset. 	           

Table 4.5: Text-to-3D Generation with SDXL (top 3) and MVDream (bottom 3).

4.4.6 Results with 6 DoF CaPE

To validate the effectiveness of the 6 DoF CaPE design, we demonstrate its performance in novel view synthesis on GSO and RTMV datasets in Tab. 4.6a and on the NeRF Synthetic dataset in Tab. 4.6c.

	Training Data	# Ref. Views	GSO-30			RTMV		
			PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
RealFusion	-	1	12.76	0.758	0.382	-	-	-
Zero123	800K	1	18.51	0.856	0.127	10.16	0.505	0.418
Zero123-XL	10M	1	18.93	0.856	0.124	10.59	0.520	0.401
EscherNet - 4 DoF	800k	1	20.24	0.884	0.095	10.56	0.518	0.410
EscherNet - 4 DoF	800k	2	22.91	0.908	0.064	12.66	0.585	0.301
EscherNet - 4 DoF	800k	3	24.09	0.918	0.052	13.59	0.611	0.258
EscherNet - 4 DoF	800k	5	25.09	0.927	0.043	14.52	0.633	0.222
EscherNet - 4 DoF	800k	10	25.90	0.935	0.036	15.55	0.657	0.185
EscherNet - 6 DoF	800k	1	20.89	0.886	0.093	12.30	0.569	0.332
EscherNet - 6 DoF	800k	2	23.92	0.917	0.057	14.18	0.618	0.252
EscherNet - 6 DoF	800k	3	25.21	0.927	0.045	15.06	0.643	0.217
EscherNet - 6 DoF	800k	5	26.59	0.937	0.036	15.71	0.663	0.190
EscherNet - 6 DoF	800k	10	27.75	0.947	0.030	16.58	0.688	0.160

(a) Novel View Synthesis Results on GSO and RTMV Datasets.

	# Ref. Views	Chamfer Dist. ↓	Volume IoU ↑
Point-E	1	0.0447	0.2503
Shape-E	1	0.0448	0.3762
One2345	1	0.0632	0.4209
One2345-XL	1	0.0667	0.4016
DreamGaussian	1	0.0605	0.3757
DreamGaussian-XL	1	0.0459	0.4531
SyncDreamer	1	0.0400	0.5220
NeuS	3	0.0366	0.5352
NeuS	5	0.0245	0.6742
NeuS	10	0.0195	0.7264
EscherNet - 4 DoF	1	0.0314	0.5974
EscherNet - 4 DoF	2	0.0215	0.6868
EscherNet - 4 DoF	3	0.0190	0.7189
EscherNet - 4 DoF	5	0.0175	0.7423
EscherNet - 4 DoF	10	0.0167	0.7478
EscherNet - 6 DoF	1	0.0274	0.6382
EscherNet - 6 DoF	2	0.0196	0.7100
EscherNet - 6 DoF	3	0.0180	0.7348
EscherNet - 6 DoF	5	0.0176	0.7392
EscherNet - 6 DoF	10	0.0160	0.7628

(b) 3D Reconstruction Results on GSO.

	# Reference Views (Less → More)							
	1	2	3	5	10	20	50	100
InstantNGP (Scene Specific Training)								
PSNR↑	10.92	12.42	14.27	18.17	22.96	24.99	26.86	27.30
SSIM↑	0.449	0.521	0.618	0.761	0.881	0.917	0.946	0.953
LPIPS↓	0.627	0.499	0.391	0.228	0.091	0.058	0.034	0.031
GaussianSplatting (Scene Specific Training)								
PSNR↑	9.44	10.78	12.87	17.09	23.04	25.34	26.98	27.11
SSIM↑	0.391	0.432	0.546	0.732	0.876	0.919	0.942	0.944
LPIPS↓	0.610	0.541	0.441	0.243	0.085	0.054	0.041	0.041
EscherNet - 4 DoF (Zero Shot Inference)								
PSNR↑	13.36	14.95	16.19	17.16	17.74	17.91	18.05	18.15
SSIM↑	0.659	0.700	0.729	0.748	0.761	0.765	0.769	0.771
LPIPS↓	0.291	0.208	0.161	0.127	0.114	0.106	0.099	0.097
EscherNet - 6 DoF (Zero Shot Inference)								
PSNR↑	13.73	15.66	16.91	17.72	18.47	18.77	19.24	19.28
SSIM↑	0.664	0.712	0.745	0.762	0.779	0.786	0.795	0.796
LPIPS↓	0.294	0.197	0.149	0.120	0.103	0.095	0.085	0.084

(c) Novel View Synthesis Results on NeRF Synthetic Dataset.

Table 4.6: **6DoF v.s. 4DoF CaPE**. 6 DoF presents a similar or improved performance than 4 DoF.

We also provide 3D reconstruction results on GSO dataset in Tab. 4.6b. It is evident that EscherNet with 6 DoF CaPE achieves comparable, and often, slightly improved results when compared to our 4 DoF CaPE design.

4.5 Discussions

Direct v.s. Autoregressive Generation EscherNet’s flexibility in handling arbitrary numbers of reference and target views offers multiple choices for view synthesis. In our experiments, we employ the straightforward direct generation to jointly generate all target views. Additionally, an alternative approach is autoregressive generation, where target views are generated sequentially, similar to text generation with autoregressive language models.

For generating a large number of target views, autoregressive generation can be significantly faster than direct generation (*e.g.* more than $20\times$ faster for generating 200 views). This efficiency gain arises from converting a quadratic inference cost into a linear inference cost in each self-attention block. However, it’s important to note that autoregressive generation may encounter a *content drifting problem* in our current design, where the generated quality gradually decreases as each newly generated view depends on previously non-perfect generated views. Autoregressive generation boasts many advantages in terms of inference efficiency and is well-suited for specific scenarios like SLAM (Simultaneous Localization and Mapping). As such, enhancing rendering quality in such a setting represents an essential avenue for future research.

Stochasticity and Consistency in Multi-View Generation We also observe that to enhance the target view synthesis quality, especially when conditioning on a limited number of reference views, introducing additional target views can be highly beneficial. These supplementary target views can either be randomly defined or duplicates with the identical target camera poses. Simultaneously generating multiple target views serves to implicitly reduce the inherent stochasticity in the diffusion process, resulting in improved generation quality and consistency. Through empirical investigations, we determine that the optimal configuration ensures a minimum of 15 target views, as highlighted in

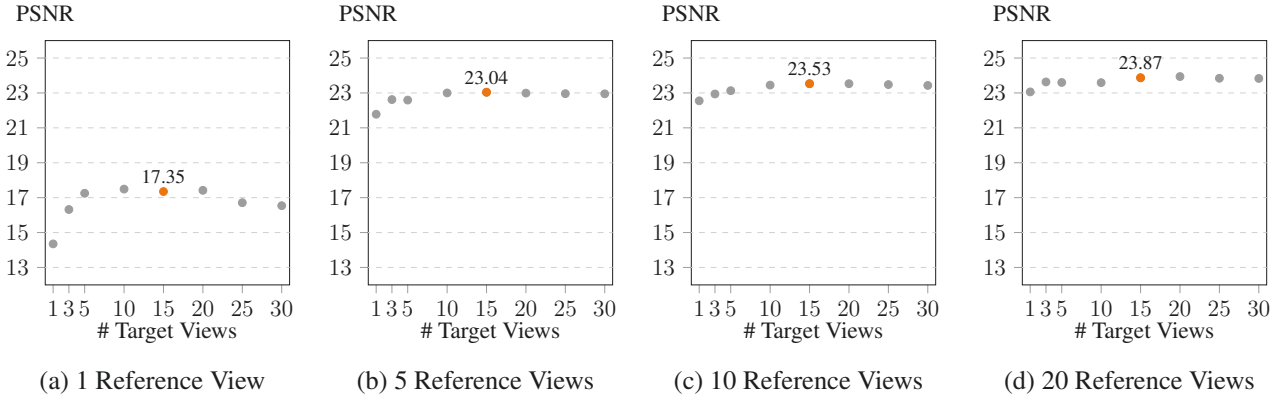


Figure 4.7: **Novel View Synthesis with a Different Number of Reference and Target Views.** We present the averaged performance of EscherNet on *one* pre-selected target view across objects in the GSO dataset. We observe a clear improvement in view synthesis quality as the number of both reference and target views increases. In this scenario, the multiple target views are essentially multiple duplicates of the initially chosen single pre-selected view, a strategy we find effective in enhancing view synthesis quality.

orange in Fig. 4.7. Beyond this threshold, any additional views yield marginal performance improvements.

Training Data Sampling Strategy We have explored various combinations of $N \in \{1, 2, 3, 4, 5\}$ reference views and $M \in \{1, 2, 3, 4, 5\}$ target views during EscherNet training. Empirically, a larger number of views demand more GPU memory and slow down training speed, while a smaller number of views may restrict the model’s ability to learn multi-view correspondences. To balance training efficiency and performance, we set our training views to $N = 3$ reference views and $M = 3$ target views for each object, a configuration that has proven effective in practice. Additionally, we adopt a random sampling approach with replacement for these 6 views, introducing the possibility of repeated images in the training views. This sampling strategy has demonstrated a slight improvement in performance compared to sampling without replacement.

Scaling with Multi-view Video EscherNet’s flexibility sets it apart from other multi-view diffusion models [LLZ⁺24, LGL⁺24] that require a set of fixed-view rendered images from 3D datasets for training. EscherNet can efficiently construct training samples using just a pair of posed images. While it can benefit from large-scale 3D datasets like [DSS⁺23, DLW⁺23], EscherNet’s adaptability extends to a broader range of posed image sources, including those directly derived from videos. Scaling

EscherNet to accommodate multiple data sources is an important direction for future research.

4.6 Conclusions

In this chapter, we have introduced EscherNet, a multi-view conditioned diffusion model designed for scalable view synthesis. Leveraging Stable Diffusion’s 2D architecture empowered by the innovative Camera Positional Embedding (CaPE), EscherNet adeptly learns implicit 3D representations from varying number of reference views, achieving consistent 3D novel view synthesis.

Limitations and Future Directions EscherNet’s flexibility in handling any number of reference views allows for autoregressive generation, similar to autoregressive language models [BMR⁺20b, CND⁺23]. While this approach significantly reduces inference time, it leads to a degraded generation quality. Additionally, EscherNet’s current capability operates within a 3 DoF setting constrained by its training dataset, which may not align with real-world scenarios, where views typically span in $SE(3)$ space. Future work will explore scaling EscherNet with 6 DoF training data with real-world scenes, striving for a more general 3D representation.

Recent Advances Building upon the ideas introduced by EscherNet, subsequent studies have evolved along two related directions. The first focuses on *multi-view diffusion synthesis*, where models such as SV3D [VYB⁺24], CAT3D [GHH⁺24], LVSM [JIT⁺25], SEVA [ZGV⁺25], and CAT4D [WGP⁺25] extend diffusion-based generation toward more coherent and scalable 3D and 4D representations. CAT3D and CAT4D integrate spatial and temporal diffusion for static and dynamic scenes, SEVA and SV3D enable flexible view and video synthesis from sparse or single-view inputs, and LVSM adopts a transformer-based architecture with Plücker ray encoding to achieve strong generalization with minimal 3D inductive bias. The second direction concerns *camera pose encoding* within attention-based frameworks, where PRoPE [LYL⁺25] extends CaPE by incorporating intrinsic parameters and systematically benchmarking different encoding strategies across tasks such as novel view synthesis, stereo depth estimation, and spatial reasoning. These developments highlight the growing role of

geometry-aware conditioning and cross-view reasoning in advancing 3D generation and understanding.

In the next chapter, we build upon the ideas introduced in EscherNet and further extend novel view synthesis to more complex, scene-level settings with arbitrary camera poses and richer semantic diversity. We propose a fully autoregressive diffusion framework with key-value caching to enable more efficient any-to-any view synthesis. This approach not only improves scalability and flexibility but also brings us closer to real-time novel view generation, with applications in AR/VR, interactive media, and world modeling.

Chapter 5

CausNVS: Autoregressive Multi-view Diffusion for 3D View Synthesis

Contents

5.1 Introduction	93
5.2 Related Work	95
5.3 CausNVS: Causal Diffusion Model for Novel View Synthesis	97
5.3.1 Training Objective	97
5.3.2 Random Per-frame Noise Conditioning	99
5.3.3 Relative Camera Pose Encoding (CaPE)	99
5.3.4 KV Caching as Spatial Memory	101
5.4 Experiments	102
5.4.1 Novel View Synthesis Results	104
5.4.2 Autoregressive v.s. Non-autoregressive	107
5.5 Conclusion	109

As discussed in the previous Chapter 4, multi-view diffusion models have shown strong potential in synthesizing novel 3D views and learning general 3D priors. However, most existing approaches



Figure 5.1: **Autoregressive Novel View Generation.** CausNVS generates target views autoregressively from an arbitrary number of input views. Albeit trained with sequences of 8 frames, it is capable of longer rollouts maintaining 3D consistency, and results continue to improve with more input views.

assume a fixed number of input and output views, which limits their applicability in dynamic or open-ended scenarios such as interactive media and continual world modeling. While EscherNet introduces a novel architecture and relative $SE(3)$ pose encoding to support multi-view to multi-view synthesis, it still requires denoising all target views simultaneously, which can be computationally inefficient and inflexible.

In this chapter, we take a step further by proposing CausNVS, an autoregressive multi-view diffusion model that flexibly handles arbitrary numbers of input and output views. Specifically, we propose CausNVS, a multi-view diffusion model in an autoregressive setting, which supports arbitrary input-output view configurations and generates views sequentially. We train CausNVS with causal masking and per-frame noise, using pairwise-relative camera pose encodings (CaPE) for precise camera control. At inference time, we combine a spatially-aware sliding-window with key-value caching and noise conditioning augmentation to mitigate drift. Our experiments demonstrate

that CausNVS supports a broad range of camera trajectories, enables flexible autoregressive novel view synthesis, and achieves consistently strong visual quality across diverse settings. Project page: <https://kxhit.github.io/CausNVS.html>. Work in this chapter was done during an internship at Google.

5.1 Introduction

World models [PHBB⁺24, AAB⁺25, Lab24, RHB⁺25, AJM⁺24, VLAF25] have emerged as a new paradigm that equips agents with internal representations of their environments, enabling them to understand, simulate, and interact with the world. A core capability of such models is spatial intelligence [Dav18], the ability to perceive 3D structure, infer spatial relationships from sparse observations, and generate photorealistic, geometrically consistent views from novel poses. In interactive or exploratory scenarios, this ability must operate sequentially and causally: as the agent moves or receives new pose queries, the system should synthesize views based only on previously accumulated information. This motivates the task of novel view synthesis (NVS), where the goal is to predict unseen views given a sparse set of posed inputs. Recent advances in multi-view diffusion models [WSL⁺24, GHH⁺24, ZGV⁺25, YAAF⁺24] have significantly improved NVS performance by incorporating large-scale pretrained 2D generative priors. However, these models are inherently non-causal, as they assume all target poses are known in advance and generate them simultaneously. This limits their ability to adapt to new queries that arrive in a streaming fashion, where future queries are fundamentally unknown.

An appealing solution is therefore to adopt a causal formulation of NVS, where views are generated autoregressively, each conditioned only on previously observed inputs and synthesized outputs. However, introducing causality into multi-view diffusion models poses two key challenges: (1) *Autoregressive Drift*. This arises from the mismatch between training and inference. The model learns from clean ground-truth views during training but must rely on its own imperfect autoregressive outputs at inference time, causing errors to accumulate and degrade future predictions. (2) *KV Caching Incompatibility*. A key advantage of autoregressive models lies in efficient inference through KV

caching [Lag23], which enables the reuse of attention computations from previously generated views. However, most existing NVS models [GHH⁺24, WSL⁺24, ZGV⁺25] rely on absolute pose encodings, such as Plücker rays [SZW19], that are tied to a global coordinate system. When the reference frame shifts (i.e., the global coordinate changes), the cached attention results become invalid and must be recomputed. Alternatively, keeping the reference frame fixed avoids recomputation, but over long rollouts, the absolute pose values grow progressively larger, eventually drifting out of training distribution, leading to a noticeable drop in generation quality.

Building on these insights, we propose CausNVS, an autoregressive multi-view diffusion model for flexible and scalable novel view synthesis. The core idea is to introduce a framewise attention layer with causal masking on top of a pretrained 2D diffusion backbone. Unlike non-causal models, our formulation enables autoregressive generation and inherently supports arbitrary input-output view configuration within a single training pass, substantially improving training efficiency.

To mitigate autoregressive drift, we apply per-frame noise conditioning during training [CMMD⁺24], which encourages the model to learn from uncertain or imperfect contexts rather than relying solely on clean ground-truth inputs. At inference time, this enables noise conditioning augmentation [HSC⁺22, VLAF25], where we can assign a small noise level to previously generated views, allowing the model to treat them as noisy inputs. This stabilizes subsequent predictions and improves robustness to accumulated errors.

To enable efficient autoregressive inference with KV caching, we adopt CaPE [KLL⁺24], a relative pose encoding that captures pairwise camera relationships independently of global coordinates. This design ensures that cached attention remains valid as the viewing window shifts, allowing for efficient sliding-window inference without recomputation or cache invalidation.

Experimental evaluation shows that CausNVS achieves consistently strong visual quality on standard fixed-view benchmarks and outperforms SoTA large-scale baselines in flexible NVS settings. It further supports stable autoregressive rollouts up to $10\times$ longer than those seen during training, demonstrating strong generalization and applicability to real-time, streaming, and open-ended view synthesis scenarios.

5.2 Related Work

Multi-view Diffusion Models Most multi-view diffusion models [GHH⁺24, WGP⁺25, WSL⁺24, WYW⁺24, TLK⁺23, YAAF⁺24, YFDB23] operate under a fixed-view setting, where the total number of views is fixed and typically consists of one or two input views along with a predefined set of target cameras. All target views are jointly denoised to enforce cross-view consistency. Some methods [SCL⁺25, YXY⁺25] further enhance geometric consistency by incorporating explicit scene geometry, such as depth maps or point cloud projections, as additional guidance. These approaches usually require extra geometry estimation pipelines [WLC⁺24] and tend to degrade when there are large pose differences between input and target views. Another line of work enhances 3D representations by leveraging pretrained image or video priors [LZH24, TRZ⁺24, PJBM23], for instance by refining NeRF [MST⁺20a] or 3D Gaussian Splatting [KKLD23]. These approaches typically rely on computationally expensive iterative optimization. To support fast inference, recent feedforward models [CXZ⁺24, CZX⁺24, XPW⁺25, JJT⁺25] directly regress novel views using transformers, but they often struggle to generalize to sparse or distant viewpoints due to the lack of generative modeling capacity.

To support more flexible view configurations, a concurrent approach, SEVA [ZGV⁺25], proposes a two-stage pipeline that clusters target views, generates anchor frames, and interpolates intermediate views using two separate diffusion models. While more adaptive than fixed-view baselines, SEVA relies on full trajectory preprocessing and handles variable-length inputs via view anchoring and padding. As a result, it still operates on fixed-size input–output tuples during inference, limiting its suitability for online or streaming scenarios.

Autoregressive Diffusion Models The use of an autoregressive formulation also aligns with recent advances in diffusion models for temporal domains. These models are well-suited to progressive or streaming generation where the output must be produced frame-by-frame rather than all at once. MAGI-1 [TJS⁺25] applies chunk-wise autoregressive denoising to videos, enabling long-horizon synthesis with prompt control. Diffusion Forcing [CMMD⁺24] introduces soft noise-level masks to unify autoregressive and parallel modes, while CausVid [YZZ⁺25] distills bidirectional video models into

causal students for low-latency autoregressive rollout. However, these approaches operate in the time domain and do not address geometric conditioning or spatial view dependencies.

More recently, works such as Game-n-Gen [VLA^{F25}] and Genie2 [PHBB⁺24] explore autoregressive generation in interactive or physically grounded environments. While closer to NVS in spirit, these methods model actions or control signals rather than camera poses. Physical dynamics such as collisions, acceleration, or other environment-induced effects prevent these actions from being reliably translated into spatial trajectories, making them difficult for direct geometric view conditioning.

3D Spatial Memory and Camera Pose Encoding As autoregressive view generation scales to longer sequences, two complementary capabilities become essential: spatial memory to retain *what was there* over time, and pose conditioning to synthesize *what is there* in each frame. The former supports long-horizon consistency, while the latter enables geometric alignment across views. This connects to recent progress in world models, which aim to synthesize coherent environments over extended horizons [AAB⁺25, PHBB⁺24, AJM⁺24, VLA^{F25}]. However, many such models lack explicit 3D pose grounding or struggle with maintaining spatial consistency due to limited context size. To address this, several works [WA24, WZH⁺25, XLZ⁺25] have proposed memory mechanisms for 3D-aware generation, often based on specialized state-space models [HS97] or external memory banks.

Meanwhile, pose conditioning has advanced for more generalizable view synthesis and precise camera control. Traditional absolute encodings, such as Plücker rays [SZW19] or direct raw extrinsics [LWVH⁺23, WYW⁺24, HXG⁺25], are tied to global coordinates and often overfit to specific view settings. EscherNet [KLL⁺24] addresses this by introducing Camera Pose Encoding (CaPE), a relative pose representation that enables more flexible input–output view configurations without retraining. While CaPE introduces a more generalizable representation, integrating spatial memory with pose-aware conditioning remains underexplored. Our work extends this direction by leveraging KV caching and pose-aware sliding-window attention to form a transformer-native memory that supports spatially consistent generation in streaming and variable-view settings.

5.3 CausNVS: Causal Diffusion Model for Novel View Synthesis

We start with a brief formulation of the autoregressive novel view synthesis (NVS) task. Given a set of N possibly discontinuous posed input views $\{(\mathbf{x}_1, \mathbf{p}_1), \dots, (\mathbf{x}_N, \mathbf{p}_N)\}$ and a sequence of M target camera poses $\{\mathbf{p}_{N+1}, \dots, \mathbf{p}_{N+M}\}$, where $\mathbf{x}_i$ is an RGB image and $\mathbf{p}_i$ is the corresponding camera pose (extrinsics and intrinsics), the goal is to generate the corresponding target views $\{\mathbf{x}_{N+1}, \dots, \mathbf{x}_{N+M}\}$. Unlike the non-autoregressive setting, where all target poses are specified in advance, the autoregressive setting assumes that target poses may arrive sequentially during inference, and thus future poses remain unknown. Consequently, each target view $\mathbf{x}_i$ must be synthesized sequentially conditioned only on the input views and the previously generated frames.

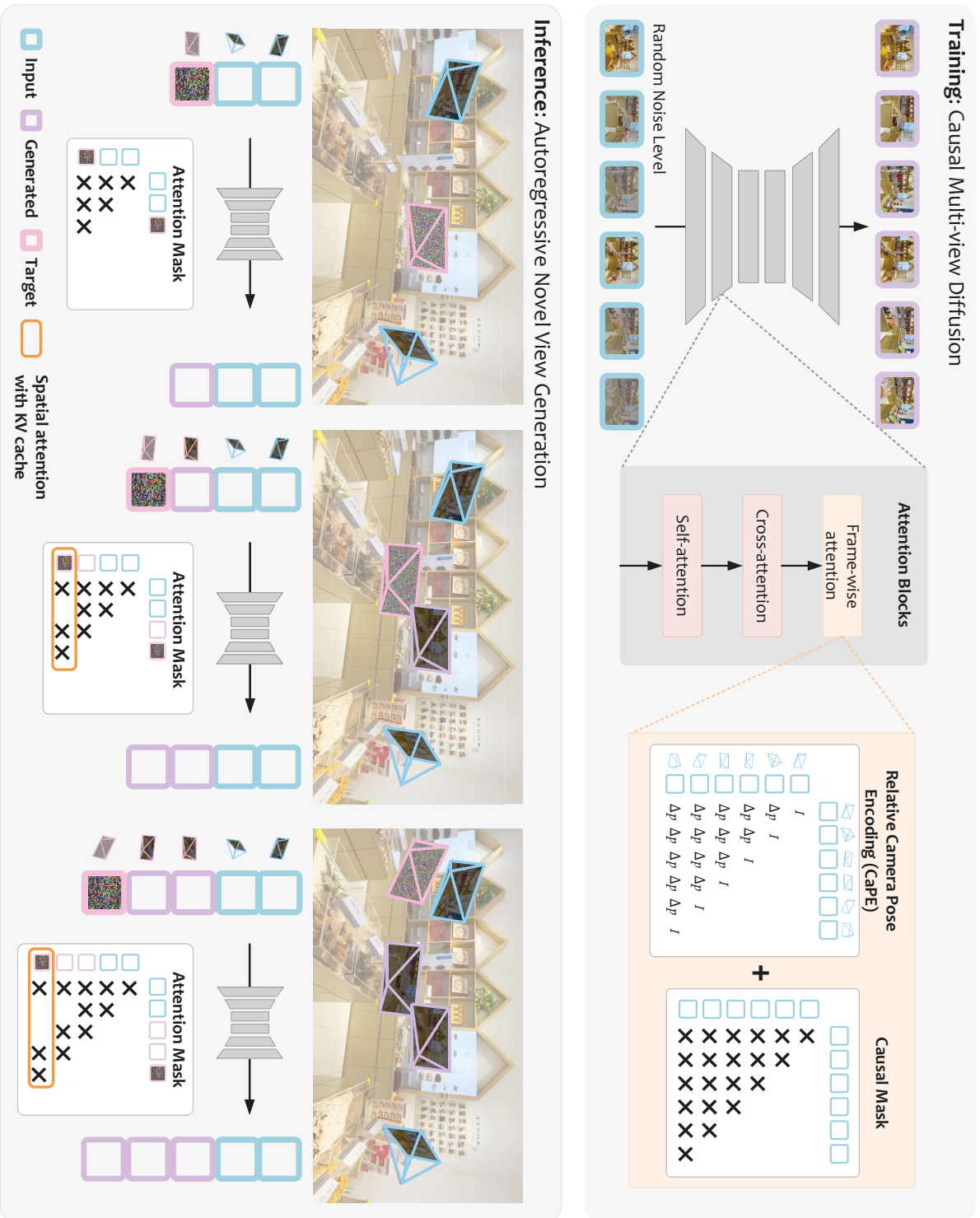
5.3.1 Training Objective

CausNVS formulates a latent diffusion model [HJA20, RBL⁺22] well-suited for the above problem setup as follows: Let $F = N + M$ denote the total number of frames in each training sequence. Each frame $\mathbf{x}_i, i = 1, \dots, F$, is encoded into a latent $\mathbf{z}_i$ using a pretrained VAE encoder [RBL⁺22] and we define the corresponding view as $\mathbf{v}_i = (\mathbf{z}_i^{t_i}, \mathbf{p}_i)$, where $\mathbf{z}_i^{t_i}$ denotes the latent $\mathbf{z}_i$ at noise level t_i and $\mathbf{p}_i$ represents camera poses. Here, we sample random independent noise levels t_i for each frame and encode them [CMMD⁺24]. This allows the model to learn denoising conditioned on partially noisy views, reducing the training-inference gap during autoregressive generation.

Further, to ensure each frame is conditioned only on past information, causal masking is applied across spatial view tokens in frame-wise attention, similar to Diffusion Forcing [CMMD⁺24] but within a UNet transformer architecture rather than a recurrent network. Under this setting, the model $\hat{\epsilon}_\theta$ is trained to predict the noise ϵ_i added to each latent $\mathbf{z}_i$, conditioned on its causal context $\mathbf{v}_{<i}$. Mathematically, the training objective is:

$$\mathcal{L}_{\text{causal}} = \mathbb{E}_{\{(\mathbf{x}_i, \mathbf{p}_i), t_i, \epsilon_i\}_{i=1}^F} \sum_{i=1}^F \|\hat{\epsilon}_\theta(\mathbf{v}_i | \mathbf{v}_{<i}) - \epsilon_i\|_2^2 \quad (5.1)$$

where $\epsilon_i \sim \mathcal{N}(0, I)$. This setup offers a key advantage: the model effectively learns a wide range



of input-target view configurations within a single training pass thanks to teacher forcing [WZ89]. Consequently, it generalizes not only to arbitrary N -to- M view predictions without requiring separate training or padding, similarly to contemporary language models, but also to varying levels of noise conditioning due to training with per-frame noise. In contrast to prior approaches [WSL⁺24, ZGV⁺25, GHH⁺24, CNC⁺23, SLS⁺24], which typically apply bidirectional denoising with shared noise levels across target views, our causal formulation with independently sampled noise allows more flexible conditioning and autoregressive generation. This also mitigates trajectory-induced biases (e.g., in 4DiM [WSL⁺24], roads tend to align with the camera path) and reduces over-reliance on the overall trajectory shape.

5.3.2 Random Per-frame Noise Conditioning

In contrast to prior video or multi-view diffusion models [BRL⁺23, WSL⁺24, GHH⁺24, KLL⁺24, ZGV⁺25] that apply a *uniform* noise level across all frames, our *independent* frame-wise noise ensures that, at inference time, conditioning on previous views with lower noise remains in-distribution even if the current frame is sampled at a higher noise level. To further improve robustness during long rollouts, we can optionally apply noise conditioning augmentation by setting small noise levels for previously generated frames, which reflects their uncertainty and helps the model reweight context and correct flawed predictions. Moreover, unlike sequential scheduled noise in Diffusion Forcing [CMMD⁺24], which assumes continuous temporal or viewpoint ordering, our setup supports arbitrary query pose orderings, a key requirement for novel view synthesis where target views may not follow a regular or continuous trajectory.

5.3.3 Relative Camera Pose Encoding (CaPE)

To leverage large-scale pre-trained image priors, we finetune a text-to-image UNet diffusion backbone similar to LDM [RBL⁺22] by inserting additional frame-wise attention layers into each attention block, as illustrated in Fig. 5.2. In these layers, we apply CaPE[KLL⁺24] to the query and key projections, encoding view-to-view geometry in a coordinate-invariant and scalable manner. Unlike

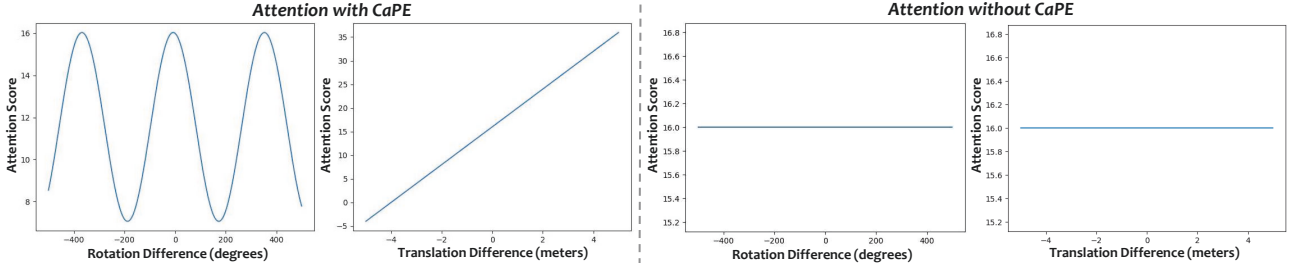


Figure 5.3: CaPE Attention Score Analysis. We analyze how attention responds to relative pose changes by initializing queries and keys randomly, and varying rotation or translation separately. With CaPE, attention scores change periodically with rotation and linearly with translation, indicating an SE(3)-aware inductive bias. Without CaPE, attention remains invariant to pose changes.

absolute pose encodings, such as Plücker rays [SZW19, GHH⁺24, WSL⁺24] and extrinsics injection [LWVH⁺23, HXG⁺25], which are tied to global coordinates and introduce extra learnable parameters, CaPE encodes only *relative* pose relationships, defined as:

$$\pi(\mathbf{v}, \mathbf{P}) = \phi(\mathbf{P})\mathbf{v}, \quad \phi(\mathbf{P}) = \mathbf{I}_{d/4} \otimes \Psi, \quad \Psi = \begin{cases} \mathbf{P} & \text{if key,} \\ \mathbf{P}^{-\top} & \text{if query,} \end{cases} \quad (5.2)$$

where $\mathbf{v} \in \mathbb{R}^d$ is the feature vector of a view, serving as a query or key token in attention, and $\mathbf{P}$ is the corresponding camera pose. $\phi(\mathbf{P}) \in \mathbb{R}^{d \times d}$ is a block-diagonal matrix constructed by tiling the 4×4 camera pose matrix Ψ .

Similar to RoPE [SLP⁺21], CaPE operates by rotating the query and key vectors in the attention computation without adding any parameters. This formulation ensures that attention scores are invariant to global coordinate changes, which is critical for causal attention and KV caching reuse in flexible NVS. In contrast, absolute pose encodings can only reuse the cache if the reference frame is preserved throughout generation; dropping it or shifting the reference would alter all subsequent pose encodings and require recomputation. As trajectories become longer, the corresponding absolute coordinates also grow larger and often exceed the normalized bounds seen during training, leading to out-of-distribution behavior. By encoding only relative pose relationships, CaPE eliminates this dependency on a fixed reference frame, allowing frames to be dropped or cached in a sliding-window manner and enabling scalable autoregressive generation with arbitrary query orderings.

Additionally, we observe that CaPE induces attention scores that vary cyclically with 360° camera



Figure 5.4: **Novel View Synthesis Results.** Visual comparison of our method and baseline models across diverse scenes and camera trajectories with input images from the DL3DV, RealEstate10K, and LLFF datasets. Each example shows the input views (N), the ground-truth target, our prediction, and a baseline result. While most baselines generate target views concurrently, our model generates them autoregressively.

rotations and linearly with translation differences, providing a strong inductive bias for 3D spatial-aware attention, as shown in Fig. 5.3. To analyze this behavior, we examine a single attention layer by randomly initializing the query and key vectors, keeping them fixed across comparisons, and computing the pre-softmax attention scores.

5.3.4 KV Caching as Spatial Memory

To enable scalable inference across long view sequences, we adopt a lightweight KV caching [Lag23] design tailored for diffusion models. A key distinction in our diffusion settings is that KV caching is required only in frame-wise attention layers, and writing to the KV cache is only necessary in the last denoising step (or the noise-augmented conditioning step), rather than across all denoising steps. This is enabled by our per-frame noise training, which supports conditioning at arbitrary noise levels during inference. In contrast, traditional video diffusion models use bidirectional attention and jointly denoise target frames, making KV caching incompatible. To further reduce attention overhead and

Method	Dataset	Re10K			LLFF		DL3DV – Long (Short)		
	N	1	2	4	1	3	1	3	6
Feed-forward									
MVSplat [CXZ ⁺ 24]		20.42	–	–	11.23	12.50	–	–	14.34
DepthSplat [XPW ⁺ 25]		20.90	–	–	12.07	12.62	9.63	12.52	15.72
Generative									
PNVS [YFDB23]		16.07	–	–	12.04	–	–	–	–
MotionCtrl [WYW ⁺ 24]		12.74	–	–	9.72	–	–	–	–
4DiM [WSL ⁺ 24]		17.08	–	–	11.58	–	–	–	–
ViewCrafter [YXY ⁺ 25]		20.43	–	–	10.53	13.52	8.97	11.50	13.78
SEVA [ZGV ⁺ 25]		12.37	16.71	19.70	14.03	19.48	13.01	15.95	17.80
CausNVS(Ours)		16.94	20.90	21.22	16.53	16.85	12.86 (13.29)	14.42 (18.32)	15.22 (18.75)

Table 5.1: **Novel View Synthesis Benchmark (PSNR $\uparrow$)**. Most baselines are designed for single-view or fixed-view conditioning and generate all target views concurrently. In contrast, our model, trained with a fixed sequence length $F = 8$, supports flexible input configurations and autoregressive generation. Despite the shorter training horizon, it achieves competitive performance across diverse datasets and view counts, while maintaining stable quality over long generation rollouts. Note: SEVA results on Re10K are obtained using the official open-source implementation.

preserve relevance, we apply a pose-aware sliding window that restricts attention to the top- K nearest views in pose space. This is possible because we train with randomized view sets, ensuring the model learns to operate under arbitrary view orderings. Together, these choices yield an implicit and efficient 3D spatial memory within the transformer: each generated view can access the most geometrically relevant prior content without recomputation. Unlike external memory systems [WZH⁺25, XLZ⁺25] that rely on separate memory buffers and explicit, lossy updates over long sequences, our cache is integrated into transformer with minimal overhead.

5.4 Experiments

We finetune our CausNVS model from a latent diffusion model similar to [RBL⁺22], following prior multi-view diffusion work [GHH⁺24, KLL⁺24]. The architecture adopts a UNet with 3 downsampling blocks, 1 middle block, and 3 upsampling blocks, operating in the VAE latent space where the spatial resolution is reduced from 256×256 to 32×32 . To support autoregressive multi-view generation, we insert frame-wise attention layers in the attention blocks with causal masking. Due to computational constraints, these layers are added only to the deeper attention blocks at resolutions 16×16 and 8×8 . These layers are zero-initialized (weights to identity, biases to zero) to preserve

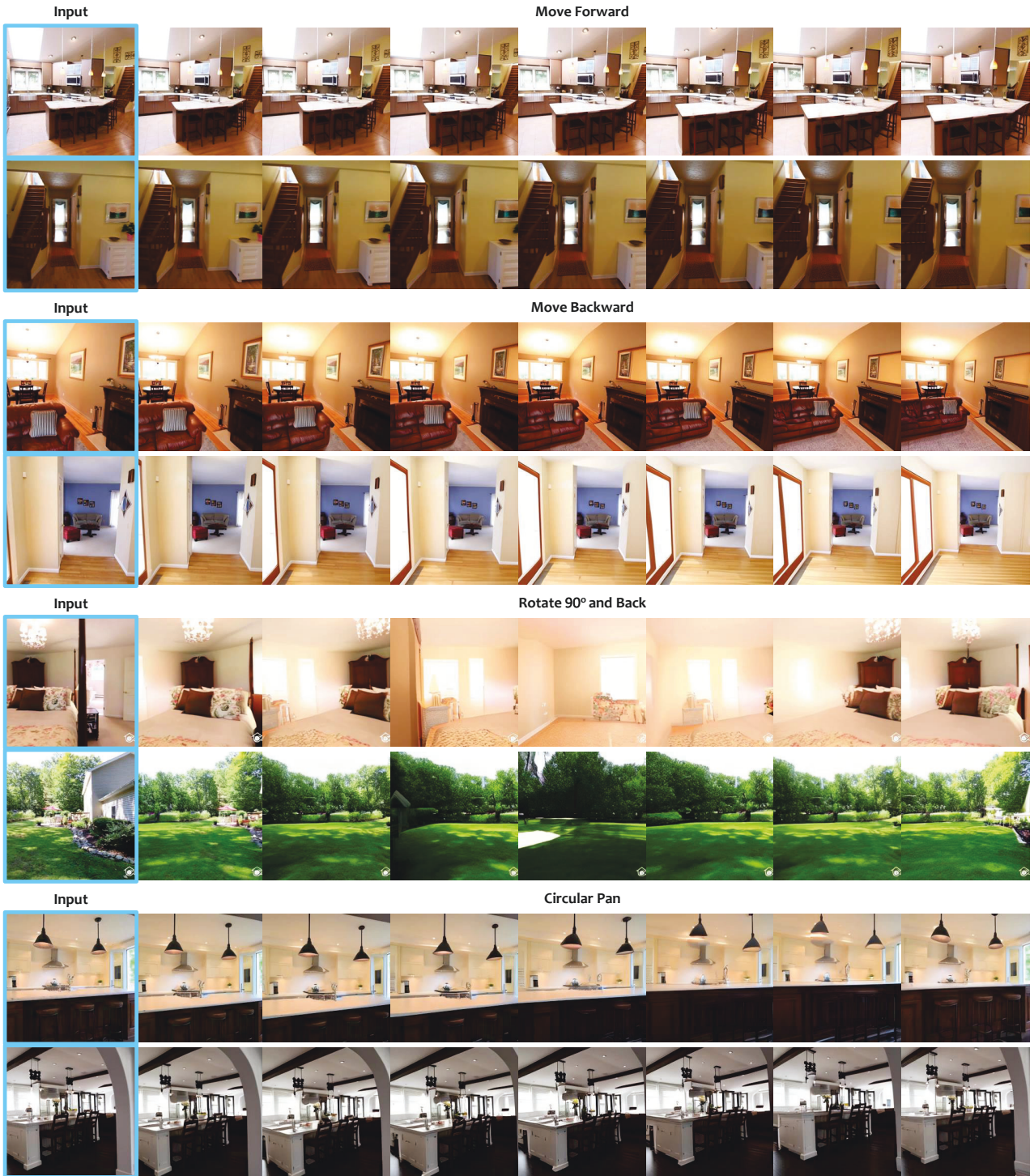


Figure 5.5: **Novel View Synthesis on Diverse Customized Trajectories.** CausNVS generalizes to diverse camera motions, including trajectories that return back, showcasing the spatial consistency.

the pretrained prior at initialization. The model has 915M parameters and is trained for 150,000 steps with a batch size of 128 on 16 TPU v5e chips. We use the Adam optimizer with a learning rate of $1e-4$ and apply a 1,000-step linear warmup. An exponential moving average (EMA) with a decay rate of 0.9999 is maintained throughout training. We train CausNVS on two public scene-level datasets

with camera pose annotations: RealEstate10K [ZTF⁺18] and DL3DV [LST⁺24]. For each training instance, a sequence of $F=8$ frames is randomly sampled from a single scene, where the selected frames may be non-contiguous.

We evaluate our method on novel view synthesis tasks across a wide range of flexible input views $N \in \{1, 2, 3, 4, 6\}$ and output views $M \in [1, 80]$, demonstrating the ability of our model to generalize across arbitrary N -to- M synthesis settings with a single trained model. We conduct evaluation following the public SEVA benchmark [ZGV⁺25] settings across Re10K [ZTF⁺18], LLFF [MSOC⁺19], and DL3DV [LST⁺24] datasets, with various number of views. We compare CausNVS to strong baselines, including generative models: SEVA [ZGV⁺25], MotionCtrl [WYW⁺24], 4DiM [WSL⁺24], ViewCrafter [YXY⁺25], PNVS [YFDB23], and feed-forward models: MVSpLat [CXZ⁺24], DepthSplat [XPW⁺25]. Among them, SEVA (1.5B) is first trained with a batch size of 1472 for $F=8$, and then further trained with a batch size of 512 for $F=21$, using a diverse mixture of object- and scene-level datasets at significantly larger scale. 4DiM is jointly trained on 3D scenes and 30 million video samples, benefiting from additional temporal supervision. ViewCrafter incorporates explicit point cloud reprojection as pose guidance, supported by a 3D foundation model [WLC⁺24]. In contrast, our model is trained with only 20K scene-level samples at a much smaller scale, while still supporting flexible N -to- M view synthesis with strong performance.

Further, in Sec. 5.4.2, we ablate the autoregressive component of CausNVS against non-autoregressive variants, highlighting the advantages of our framework in flexible N -to- M novel view synthesis with unified single-model training, as well as the efficiency and 3D consistency under novel trajectories enabled by spatial attention windows with KV caching as spatial memory.

5.4.1 Novel View Synthesis Results

Novel View Synthesis Benchmark As shown in Tab. 5.1 and Fig. 5.4, CausNVS achieves competitive performance across diverse scenes and improves consistently with more input views in the SEVA benchmark [ZGV⁺25]. CausNVS generates views autoregressively and supports flexible N -to- M synthesis within a single unified model, while all other baselines need to generate novel views simul-

Re10K N	PSNR $\uparrow$			SSIM $\uparrow$			LPIPS $\downarrow$		
	1	2	4	1	2	4	1	2	4
4DiM	17.08	-	-	0.463	-	-	0.302	-	-
SEVA	12.37	16.71	19.70	0.469	0.590	0.673	0.577	0.346	0.221
CausNVS(Ours)	16.94	20.90	21.22	0.466	0.690	0.718	0.279	0.119	0.095

LLFF N	PSNR $\uparrow$		SSIM $\uparrow$		LPIPS $\downarrow$	
	1	3	1	3	1	3
4DiM	11.58	-	0.141	-	0.542	-
SEVA	14.03	19.48	0.384	0.602	0.389	0.181
CausNVS(Ours)	16.53	16.85	0.272	0.336	0.336	0.222

Table 5.2: **Novel view synthesis metrics on Re10K and LLFF.** Note: SEVA results on Re10K are obtained by running the official public code. No scale sweeping is applied at $N=1$.

taneously with fixed view counts. CausNVS maintains consistent visual quality across varying inputs and wide pose differences, where baseline results often exhibit blur or distortion. We provide more metrics (SSIM, LPIPS) for novel view synthesis evaluation across main baselines (4DiM, SEVA) in Tab. 5.2 and more visualization:

- **Customized Pose Trajectories (Fig. 5.5):** CausNVS generates consistent and coherent outputs under diverse camera motion patterns such as forward/backward motion, rotations and circular pans trajectories.
- **Effect of Increasing Input Views on DL3DV (Fig. 5.7–5.8):** We observe clear benefits from more input views, yielding results that are more 3D-consistent and semantically faithful to observations.
- **Comparisons on Re10K with Baselines (Fig. 5.9–5.10):** CausNVS achieves superior pose control and scene consistency compared to strong baselines.

Feed-forward methods MVSplat [CXZ⁺24] and DepthSplat [XPW⁺25] perform well when the input views are densely sampled with small baselines, but their performance deteriorates significantly under wide view gaps. Due to the lack of generative capability, they typically fail to extrapolate novel views beyond the observed ones. Generative models [YFDB23, WYW⁺24, WSL⁺24, YXY⁺25, ZGV⁺25]

often assume a fixed single-view input and struggle with variable-length configurations. Methods like 4DiM [WSL⁺24] address this limitation via multi-view fine-tuning, but at the cost of additional training overhead. ViewCrafter [YXY⁺25] incorporates an explicit 3D point cloud estimator as geometric guidance, yet its performance degrades when the target view is far from the inputs. SEVA [ZGV⁺25], developed concurrently with our work, supports varying numbers of input views but still assumes fixed-size input-output pairs. It relies on ad hoc heuristics such as view anchoring, chunking, and padding, combined with multiple forward passes, to approximate variable-length conditioning. In contrast, our causal model supports autoregressive generation with arbitrary numbers of input views, without requiring retraining, architectural modifications, or complex sampling strategies at inference.

Single View Scale Ambiguity Scale ambiguity remains a fundamental challenge in single-view novel view synthesis. Some baselines address this via scale sweeping: 4DiM [WSL⁺24] uses pre-processed depth estimates to recover metric scale, while SEVA [ZGV⁺25] normalizes camera scale by sweeping from 0.1 to 2.0 and selecting the best-performing result per scene. For example, SEVA’s PSNR on Re10K improves from 12.37 to 17.99 when using sweeping. However, we intentionally avoid such scale tuning during inference. First, sweeping requires access to ground truth to select the optimal scale, which limits generalization and realism. Second, our method is designed for flexible N -to- M view synthesis in generative 3D settings, where single-view is an ambiguous special case rather than the primary focus. Third, we prioritize keeping the inference pipeline simple and consistent across varying input conditions, without hand-tuned parameters. Despite not relying on scale sweeping, our model performs competitively in the single-view setting and consistently improves as more views are provided, effectively resolving scale ambiguity via multi-view cues.

Autoregressive Generation Drift To evaluate the stability of our autoregressive rollout, we report results on both the short and long sequence splits of DL3DV, while keeping the input views fixed. The short sequence is a random subset of the long one, covering the same spatial region but with fewer target views. This setup enables us to isolate the effect of autoregressive drift from different input views. As shown in Tab. 5.1, performance on short sequences consistently improves with more input views. On long sequences, where the number of target views M ranges from $4\times$ to $10\times$ the training

Model	Infer	N/F	2	4	8	32	64
Causal	AR	1	16.56 / –	17.25 / 81.25	18.08 / 95.83	17.44 / 99.79	16.98 / 99.38
		2	–	19.52 / 93.75	20.40 / 93.75	21.10 / 99.78	19.31 / 99.49
		4	–	–	23.27 / 91.67	24.55 / 100.0	23.35 / 99.89
		16	–	–	–	28.13 / 100.0	27.64 / 99.87
Non-Causal	Parallel	1	9.84 / –	13.15 / 34.38	17.52 / 92.71	19.27 / 100.0	18.15 / 100.0
		2	–	18.38 / 53.33	20.61 / 97.50	21.86 / 100.0	21.42 / 99.90
		4	–	–	23.97 / 93.75	25.12 / 100.0	24.29 / 100.0
		16	–	–	–	28.56 / 99.17	27.64 / 99.47
	AR	2	–	–	13.62 / 86.77	–	–
					–	–	–

Table 5.3: **N-to-M Novel View Synthesis Ablation (PSNR $\uparrow$ / TSED $\uparrow$)**. AR denotes autoregressive inference, and Parallel denotes concurrent inference. We evaluate causal and non-causal models with varying numbers of input views N and total frames F on the Re10K validation set. The causal model maintains consistent performance across different configurations with stable autoregressive rollout. In contrast, the non-causal model performs poorly when the evaluation sequence is shorter than its training setup, even under parallel denoising, and quickly drifts when used autoregressively.

length ($F = 8$), our model exhibits some degradation compared to the short case, reflecting the inherent challenges of long-context generation. However, it remains competitive with most baselines. SEVA benefits significantly from its longer training sequence length ($F = 21$) and parallel decoding. We believe scaling to longer sequences would further improve long-context performance.

5.4.2 Autoregressive v.s. Non-autoregressive

N-to-M Novel View Synthesis Ablation We conduct ablation studies on RealEstate10K [ZTF⁺18] dataset, comparing our CausNVS model to a model trained identically albeit removing causality, following established protocols [WSL⁺24, GHH⁺24, ZGV⁺25]. Both models share the same architecture and training configuration to ensure a fair comparison. We evaluate using two complementary metrics: PSNR, reflecting image quality, and TSED [YFDB23, WSL⁺24], a 3D structural alignment measurement between generated and ground-truth views, using a threshold of 2. As shown in Tab. 5.3, the causal model generalizes reliably from a single view to $8\times$ training length, and improves consistently with more input views. In comparison, the non-causal model performs well only under conditions close to its training setup and degrades significantly when evaluated with different sequence lengths. In the one-to-one generation case shown in Fig. 5.6, it produces unstable results and exhibits drift across frames. This is consistent with observations from previous

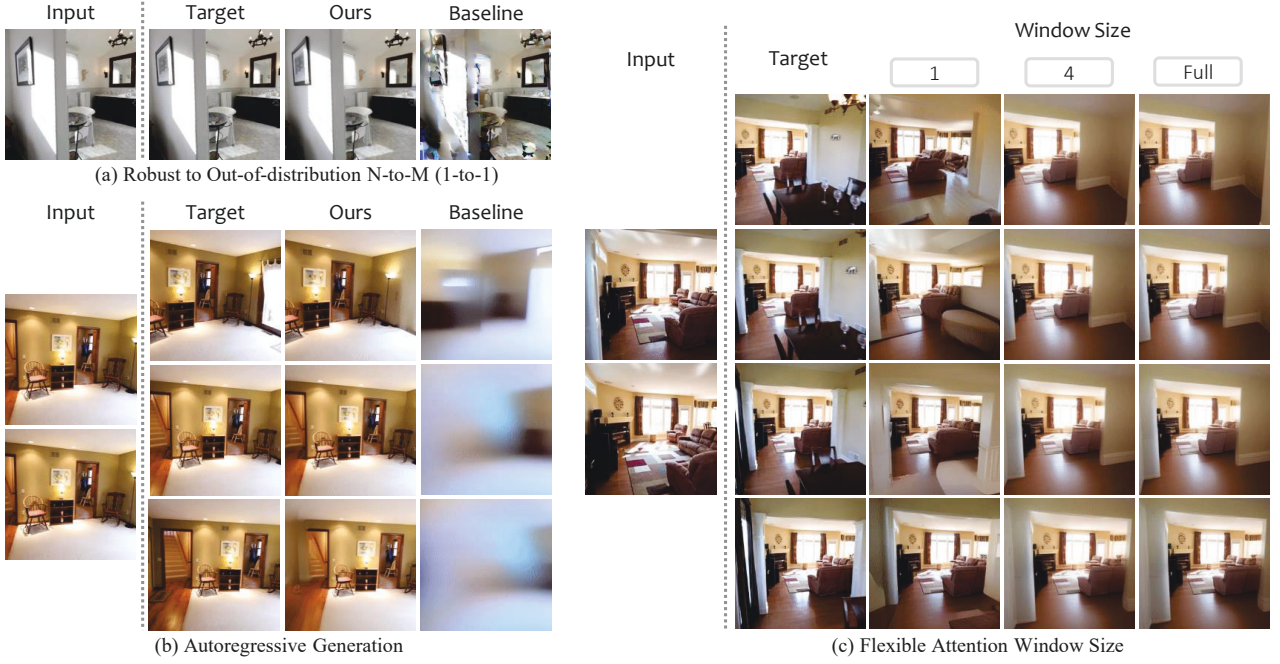


Figure 5.6: **Examples of Key Properties of CausNVS.** Through qualitative examples, we demonstrate key properties of CausNVS compared to non-causal baselines. (a) Although trained with a fixed sequence length F , causal masking enables supervision for all sequence lengths up to F . This leads to robust generalization, including 1-to-1 settings where non-causal models often fail. (b) Under autoregressive rollout, our model maintains visual stability, while the non-causal baseline quickly drifts and produces blurry results. (c) CausNVS supports a configurable attention window size at inference time. With window size 1, attention is limited to input views, leading to inconsistency across generated frames. With window size 4, CausNVS already achieves comparable results to full attention at a fraction of the inference FLOPS.

work [GHH⁺24, KLL⁺24, ZGV⁺25], which report that non-causal models often require jointly denoising multiple padded frames to stabilize outputs. Such strategies introduce redundant computation and limit flexibility at inference time.

Spatial Attention Window and Memory CausNVS supports a flexible spatial attention window that dynamically selects relevant views at inference time, enabling efficient context aggregation without incurring full attention cost. This pose-aware sliding strategy ensures broad spatial coverage even under limited attention window, and achieves results comparable to global attention with significantly reduced FLOPS (Fig. 5.6c). By combining this with lightweight KV caching, CausNVS retains information from previously generated views without recomputation, allowing context to persist implicitly across frames.

We further demonstrate in Fig. 5.5 that this mechanism enables CausNVS to maintain spatial con-

sistency under diverse, user-defined camera trajectories, including those with revisited views. Each generated view can retrieve geometrically relevant prior content, preserving both structure and appearance across varying and revisited viewpoints. In practice, this combination of spatial windowing and KV caching serves as an implicit and efficient spatial memory, ensuring coherence across arbitrary camera motions.

5.5 Conclusion

We have presented CausNVS, a causal multi-view diffusion model that supports autoregressive generation with flexible numbers of input and output views. The model is designed with causal masking, noise-level conditioning, and relative pose encoding (CaPE) embedded within a unified decoder-only architecture. This design supports efficient pose-aware spatial memory and KV caching at inference time, enabling robust and efficient novel view synthesis suitable for streaming and interactive applications such as AR/VR, world modeling, and generative content creation.

Limitations and Future Directions While effective, our model relies on multi-step denoising, limiting its real-time applicability [SME21]. Future work could explore faster generation via consistency training [SDCS23] or distillation [YZZ⁺25, XXK⁺24, SLBR24]. Scaling to longer sequence and more diverse datasets (e.g., objects, videos) would improve generalization with stronger priors [WSL⁺24, WGP⁺25, ZGV⁺25]. Additionally, integrating audio, language, and action signals into the model could lead toward fully multimodal [Tea24, ZYB⁺25, XMB⁺25, KPK⁺24] world models with spatial grounding and controllable rollout in complex environments.

Broader Impacts Our method may benefit applications in AR/VR, world modeling, and 3D content creation, but it also carries potential risks such as misuse in synthetic media or simulation bias. We advocate responsible model release and transparent benchmarks to support safe deployment.



Figure 5.7: Novel View Synthesis on DL3DV with Increasing Input Views. [Part 1]

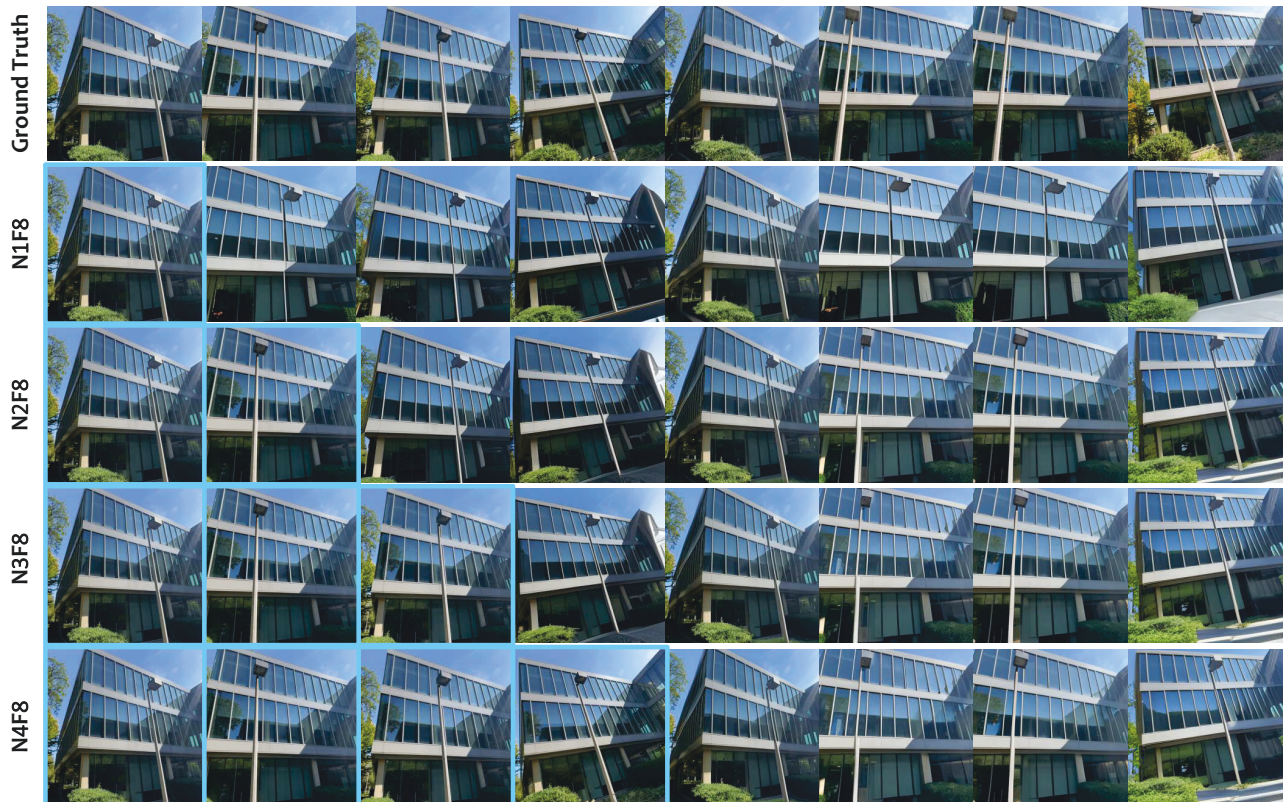


Figure 5.8: Novel View Synthesis on DL3DV with Increasing Input Views. [Part 2]

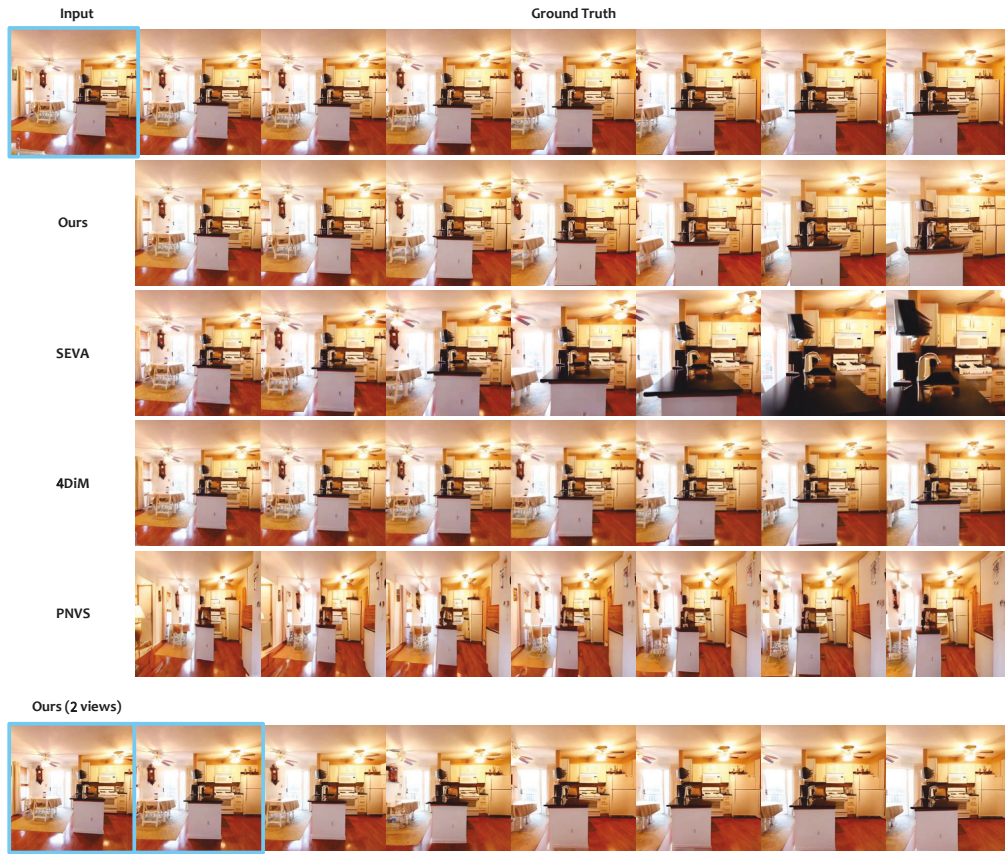


Figure 5.9: Novel View Synthesis on Re10K Compared with Baselines. [Part 1]

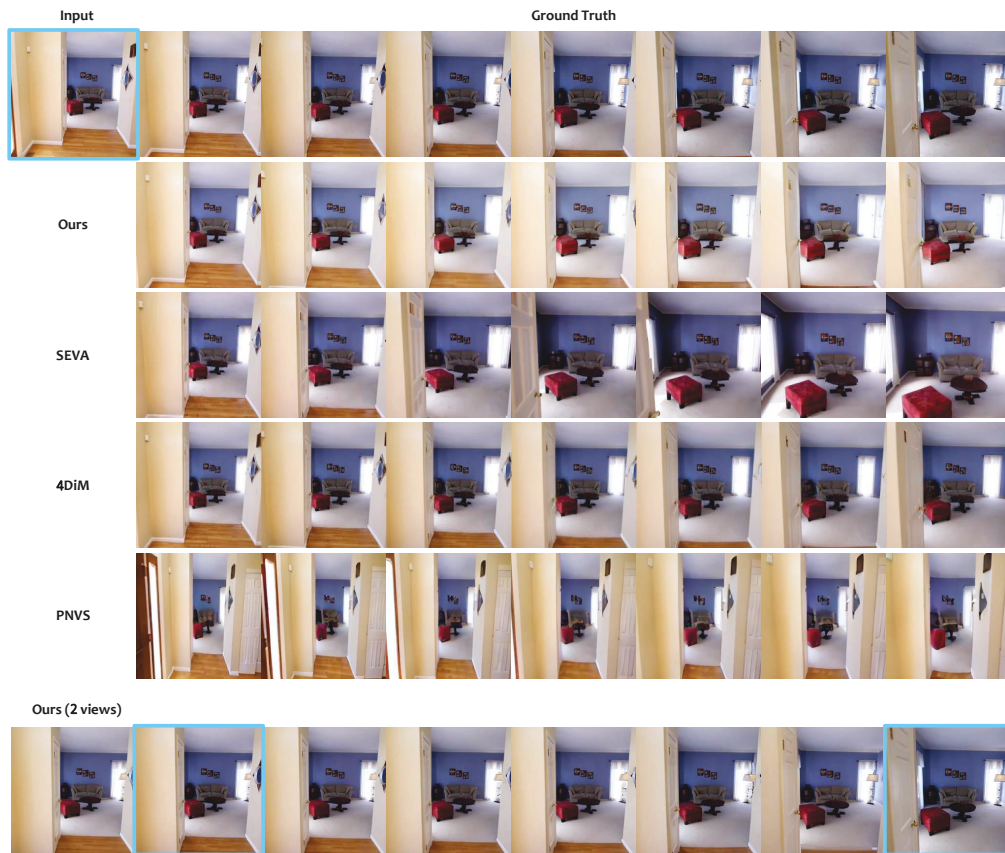


Figure 5.10: Novel View Synthesis on Re10K Compared with Baselines. [Part 2]

Chapter 6

Conclusions and Future Work

This thesis presents a coherent progression of methods for learning neural 3D scene representations and synthesising novel views from posed images. Starting from scene-specific object-level mapping, and advancing through generalisable multi-view synthesis, to fully autoregressive scene-level generation, each chapter addresses a key challenge in scaling 3D perception and generation to more flexible, efficient, and real-world-ready systems. Together, the three core contributions form a unified framework for learning to perceive, represent, and generate 3D scenes using neural networks and diffusion models.

6.1 Summary of Contributions

Chapter 3 introduces vMAP, a real-time dense SLAM system that models each object in a scene with a compact and independent MLP-based neural field. Unlike prior methods that rely on monolithic scene representations or 3D shape priors, vMAP enables per-object modelling with watertight reconstruction, achieved without any prior shape knowledge. Through vectorised training, it can optimise dozens of objects simultaneously on a single GPU, achieving both efficiency and flexibility. The model supports recomposition, dynamic updates, and tracking, making it suitable for robotics and interactive spatial applications.

Chapter 4 focuses on generalisation across scenes, proposing EscherNet, a multi-view conditioned diffusion model for novel view synthesis. EscherNet decouples 3D representation from global coordinates and scene-specific optimisations. It operates entirely in image space and learns to reason about geometry from relative camera transformations using Camera Positional Embedding (CaPE). Despite being trained on a fixed number of views, it generalises to arbitrary input and output view configurations and achieves state-of-the-art performance across novel view synthesis and 3D reconstruction benchmarks. EscherNet demonstrates the effectiveness of large-scale pretraining for learning implicit 3D priors.

In Chapter 5, we propose CausNVS, an autoregressive multi-view diffusion model designed for open-ended and streaming scenarios. CausNVS uses causal masking across view tokens to support variable-length input and output view sequences, while key-value caching and relative pose-aware attention provide scalable spatial memory. During inference, the model generates novel views sequentially, reusing attention states for efficiency. It can roll out long view sequences with high geometric consistency and robustness, enabling interactive applications such as real-time scene exploration, world models, and generative simulation.

6.2 Future Work

Scalable 3D Representation A key challenge ahead lies in designing 3D representations that balance geometric expressiveness with scalability. While neural fields offer rich geometry, they require dense supervision and are costly to optimise. Future work could explore representations such as neural primitives [MBD24], feed-forward Gaussian splats [XPW⁺25, SZS⁺25, SMK25], or hybrid implicit-explicit formats [SCP⁺23, ZYG⁺24] that can preserve multiview consistency and scale efficiently with transformer-based architectures. These representations should support sparse input, learnable priors, and rapid adaptation across scenes.

3D and 4D Foundation Models As vision-language models evolve into general-purpose foundation models, a parallel opportunity emerges in the domain of 3D and 4D spatial understanding. Recent

works such as DUST3R, MAST3R, and VGGT [WLC⁺24, LCR24, WCK⁺25] have shown strong performance in pose estimation and 3D reconstruction using dense point maps as supervision. In parallel, another line of research [JJT⁺25, GHH⁺24, WGP⁺25], including our own, explores novel view synthesis from posed images.

A promising future direction is to develop large-scale spatial foundation models that unify multiple 3D tasks—including pose estimation, reconstruction, and view synthesis—within a single architecture. These models can be trained with multiview self-supervised learning objectives such as masked view prediction [HCX⁺22, WLL⁺22, WLL⁺23, WMK⁺25], enabling a gradual transition from strongly supervised signals like depth and point correspondences to weaker but more scalable supervision, such as posed images or even raw videos [JTW⁺25]. By leveraging large-scale multiview and video datasets, such models could serve as general-purpose spatial priors, transferable across a wide range of downstream 3D and 4D tasks.

SE(3)-Equivariant Positional Encoding Transformer models for 3D tasks benefit significantly from positional encodings that capture the geometric structure of Euclidean space. While our proposed CaPE [KLL⁺24] introduces a relative, camera-aware encoding scheme that improves view consistency and flexibility, there remains substantial room for further exploration of SE(3)-equivariant mechanisms that provide stronger geometric inductive bias in attention.

In the language domain, LLMs leverage rotary positional embeddings (RoPE) [SLP⁺21] to support stable attention across extremely long contexts, with effective window sizes exceeding 100k tokens. In contrast, 3D vision models often adopt absolute position encodings such as Plücker ray embeddings, which are tied to a global coordinate frame and struggle to generalise across larger or shifting spatial domains.

Future research may explore extending rotary-style embeddings to 3D settings, enabling spatially aware attention that respects relative pose transformations [KLL⁺24, MJWG24]. Incorporating intrinsic camera parameters [LYL⁺25] into the encoding may further enhance generalisation across diverse view geometries and scene configurations, contributing to more robust and scalable 3D transformers.

Spatial Memory and Continual Learning As models begin to operate over long sequences and dynamic environments, effective memory becomes essential. Traditional recurrent memory modules like LSTMs [HS97] compress past information into abstract hidden states but are often inefficient and unstable. A promising alternative is to adapt cache-based [TDBM22] or retrieval-augmented generation (RAG) [LPP⁺20] from language models to 3D vision, allowing the model to selectively access prior views or scene elements based on spatial context [CYZ⁺25]. Research is also needed to enable continual learning, where models can accumulate knowledge across time without forgetting previously learned structures.

Real-time Generative Reconstruction Diffusion models offer high-quality generation but remain computationally expensive due to their iterative nature. Future work could investigate shortcut diffusion models [FHLA25], flow-based generation [GDB⁺25], and adaptive denoising strategies that reduce steps without sacrificing quality [CMMD⁺24]. For example, RGB view synthesis may require more denoising steps due to its high-frequency content, while depth or semantic maps may be completed in fewer steps. Task-aware inference scheduling, consistency training, and efficient autoregressive caching [KWS⁺25, HLH⁺25] can bring diffusion models closer to real-time deployment.

From Spatial AI to Embodied AI A key future direction is to bridge the gap between spatial AI and embodied AI, enabling agents to perceive, reason, and act in the physical world. Current end-to-end embodied systems often rely on fine-tuning large 2D vision-language models (VLM) for action prediction (VLA) [KPK⁺24, BBD⁺24, IBB⁺25], without incorporating explicit 3D spatial reasoning or verification. However, spatial understanding serves as a crucial intermediate representation that connects 2D perception and language with real-world physical interaction.

Developing robust 3D priors can significantly reduce the amount of data required in action space by providing structured geometric knowledge and improving sample efficiency. This is particularly important given that most existing behavior cloning and imitation learning approaches depend on expensive, large-scale offline action datasets. By first grounding models in spatial understanding, and then extending to visual-language-action learning, future systems may achieve more data-efficient, generalisable, and reliable embodied intelligence.

Bibliography

- [AAB⁺25] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. 93, 96
- [ACDS22] Jad Abou-Chakra, Feras Dayoub, and Niko Sünderhauf. Implicit object mapping with noisy data. *arXiv preprint arXiv:2204.10516*, 2022. 41
- [AHG⁺19] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 16
- [AJM⁺24] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 93, 96
- [BBB⁺25] Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttmore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi,

- Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov, Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A new frontier for world models. <https://deepmind.google/discover/blog/genie-3-a-new-frontier-for-world-models/>, 2025. 9, 35
- [BBD⁺24] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0 : A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 115
- [BDE⁺24] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024. 8, 35
- [BDK⁺23] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 27
- [BDPD⁺92] Peter F Brown, Vincent J Della Pietra, Peter V Desouza, Jennifer C Lai, and Robert L Mercer. Class-based n-gram models of natural language. *Computational linguistics*, 1992. 31
- [BMH⁺22] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan

- Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2022. 31
- [BMR⁺20a] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 32
- [BMR⁺20b] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 89
- [BMV⁺22] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 55
- [BPC20] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020. 30
- [BRL⁺23] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 27, 28, 99
- [CER⁺21] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics (T-RO)*, 2021. 18, 47, 52
- [CFG⁺15] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 66

- [CGC⁺23] Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 68
- [CGRS19] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 30
- [CLD⁺21] Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 31
- [CMMD⁺24] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 33, 94, 95, 97, 99, 115
- [CNC⁺23] Eric R Chan, Koki Nagano, Matthew A Chan, Alexander W Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3d-aware diffusion models. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 99
- [CND⁺23] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research (JMLR)*, 2023. 89
- [CRC⁺20] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020. 32

- [CXG⁺22] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 66
- [CXZ⁺24] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 95, 102, 104, 105
- [CYZ⁺25] Shengqu Cai, Ceyuan Yang, Lvmin Zhang, Yuwei Guo, Junfei Xiao, Ziyang Yang, Yinghao Xu, Zhenheng Yang, Alan Yuille, Leonidas Guibas, et al. Mixture of contexts for long video generation. *arXiv preprint arXiv:2508.21058*, 2025. 115
- [CZJ⁺22] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 32, 33
- [CZX⁺24] Yuedong Chen, Chuanxia Zheng, Haofei Xu, Bohan Zhuang, Andrea Vedaldi, Tat-Jen Cham, and Jianfei Cai. Mvsplat360: Feed-forward 360 scene synthesis from sparse views. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 95
- [Dav18] Andrew J Davison. Futuremapping: The computational structure of spatial ai systems. *arXiv preprint arXiv:1803.11288*, 2018. 2, 93
- [DCS⁺17] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 46, 50
- [DFK⁺22] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2022. 77, 78

- [DJQ⁺23] Congyue Deng, Chiyu Jiang, Charles R Qi, Xinchun Yan, Yin Zhou, Leonidas Guibas, Dragomir Anguelov, et al. Nerdi: Single-view nerf synthesis with language-guided diffusion as general image priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 68
- [DLW⁺23] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 68, 88
- [DLZR22] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 67
- [DNJ21] Thomas Davies, Derek Nowrouzezahrai, and Alec Jacobson. On the effectiveness of weight-encoded neural implicit 3d shapes. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. 41
- [DNZ⁺17] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (TOG)*, 2017. 18, 39
- [DRMS07] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2007. 4, 18
- [DSS⁺23] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 68, 76, 88
- [EHE⁺12] Felix Endres, Jürgen Hess, Nikolas Engelhard, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. An Evaluation of the RGB-D SLAM System. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012. 46

- [EKC17] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2017. 18
- [EMS⁺23] Ziya Erkoç, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 68
- [ESC14] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. 18
- [FHLA25] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. 115
- [FKYT⁺22] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5
- [GDB⁺25] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 115
- [GDG19] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 47
- [GHH⁺24] Ruiqi Gao*, Aleksander Holynski*, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul P. Srinivasan, Jonathan T. Barron, and Ben Poole*. Cat3d: Create anything in 3d with multi-view diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 28, 89, 93, 94, 95, 99, 100, 102, 107, 108, 114

- [GKJ⁺21] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 64, 66
- [GLT⁺20] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020. 31
- [HCX⁺22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 114
- [HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 7, 23, 67, 97
- [HLBN20] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 8, 35
- [HLD⁺19] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 2019. 59
- [HLH⁺25] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 33, 115
- [HLNB21] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 8, 35
- [HPBL23] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. 8, 35

- [HR21] He Horace and Zou Richard. functorch: Jax-like composable function transforms for pytorch. <https://github.com/pytorch/functorch>, 2021. 43
- [HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997. 31, 96, 115
- [HS18] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. 8, 35
- [HS22] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 26
- [HSC⁺22] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research (JMLR)*, 2022. 94
- [HSG⁺22] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 27
- [HSM⁺21] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 66
- [Hua25] Xun Huang. Towards video world models. https://www.xunhuang.me/blogs/world_model.html, 2025. 35
- [HWP⁺24] Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. Lm-infinite: Zero-shot extreme length generalization for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2024. 71
- [HXG⁺25] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation.

- In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. 96, 100
- [IBB⁺25] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Conference on Robot Learning (CoRL)*, 2025. 115
- [JA21] Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 41
- [JJT⁺25] Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. 89, 95, 114
- [JN23] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 68, 78, 81
- [JTA21] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 67
- [JTW⁺25] Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, et al. Rayzer: A self-supervised large view synthesis model. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2025. 114
- [KAAL22] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 25
- [KKJ⁺24] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat track & map 3d

- gaussians for dense rgb-d slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 19
- [KKLD23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 2023. 5, 15, 16, 19, 64, 66, 77, 79, 81, 95
- [KLL⁺24] Xin Kong, Shikun Liu, Xiaoyang Lyu, Marwan Taher, Xiaojuan Qi, and Andrew J Davison. Eschernet: A generative model for scalable view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 10, 94, 96, 99, 102, 108, 114
- [KLTD23] Xin Kong, Shikun Liu, Marwan Taher, and Andrew J Davison. vmap: Vectorised object mapping for neural field slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 10
- [KM09] Georg Klein and David Murray. Parallel tracking and mapping on a camera phone. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2009. 18
- [KPK⁺24] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024. 109, 115
- [KW14] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014. 27
- [KWS⁺25] Xin Kong, Daniel Watson, Yannick Strümpfer, Michael Niemeyer, and Federico Tombari. Causnvs: Autoregressive multi-view diffusion for flexible 3d novel view synthesis. *arXiv preprint arXiv:2509.06579*, 2025. 10, 115
- [KYZ⁺20] Xin Kong, Xuemeng Yang, Guangyao Zhai, Xiangrui Zhao, Xianfang Zeng, Mengmeng Wang, Yong Liu, Wanlong Li, and Feng Wen. Semantic graph based place

- recognition for 3d point clouds. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2020. 40
- [Lab24] World Labs. Generating worlds. <https://www.worldlabs.ai/blog>, 2024. 93
- [Lag23] João Lages. Kv caching explained. <https://medium.com/@joaolages/kv-caching-explained-276520203249>, 2023. 94, 101
- [LC98] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*. 1998. 59
- [LCBH⁺23] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. 26
- [LCR24] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 19, 114
- [LDR⁺22] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 64
- [LDW⁺25] Yuzheng Liu, Siyan Dong, Shuzhe Wang, Yingda Yin, Yanchao Yang, Qingnan Fan, and Baoquan Chen. Slam3r: Real-time dense scene reconstruction from monocular rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 20
- [LGL23] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. 26

- [LGL⁺24] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 65, 68, 79, 88
- [LGT⁺23] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 67
- [LH19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 76
- [LJD19] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 58
- [LLW⁺21] Xiaoyang Lyu, Liang Liu, Mengmeng Wang, Xin Kong, Lina Liu, Yong Liu, Xinxin Chen, and Yi Yuan. Hr-depth: High resolution self-supervised monocular depth estimation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2021. 60
- [LLY⁺22] Guanglin Li, Yifeng Li, Zhichao Ye, Qihang Zhang, Tao Kong, Zhaopeng Cui, and Guofeng Zhang. Generative category-level shape and pose estimation with semantic primitives. In *Conference on Robot Learning (CoRL)*, 2022. 40
- [LLZ⁺24] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 28, 65, 68, 70, 72, 77, 79, 81, 88
- [LPP⁺20] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al.

- Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 115
- [LST⁺24] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. D13dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 104
- [LWVH⁺23] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 6, 7, 28, 29, 65, 68, 70, 72, 76, 78, 79, 96, 100
- [LXJ⁺23] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 65, 68, 79, 81
- [LYL⁺25] Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 89, 114
- [LZH24] Xi Liu, Chaoyi Zhou, and Siyu Huang. 3dgs-enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 95
- [MAMT15] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics (T-RO)*, 2015. 4, 18
- [MAT17] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics (T-RO)*, 2017. 18

- [MBD24] Kirill Mazur, Gwangbin Bae, and Andrew J Davison. Superprimitive: Scene reconstruction at a primitive level. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 16, 113
- [MCB⁺18] John McCormac, Ronald Clark, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Fusion++: Volumetric object-level slam. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2018. 19, 40, 59
- [MCSM18] Alexander Majercik, Cyril Crassin, Peter Shirley, and Morgan McGuire. A ray-box intersection algorithm and efficient dynamic voxel rendering. *Journal of Computer Graphics Techniques (JCGT)*, 2018. 46
- [MDD25] Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 19
- [MESK22] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 2022. 5, 40, 60, 64, 66, 77, 79
- [MHDL17] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semantic-fusion: Dense 3d semantic mapping with convolutional neural networks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 19
- [MJWG24] Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. Gta: A geometry-aware attention mechanism for multi-view transformers. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 114
- [MKB⁺10] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, 2010. 31
- [MKLRV23] Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Realfusion: 360deg reconstruction of any object from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 68, 79

- [MLC25] Dominic Maggio, Hyungtae Lim, and Luca Carlone. Vggt-slam: Dense rgb slam optimized on the $sl(4)$ manifold. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 20
- [MMKD24] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 19
- [MON⁺19] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4, 40, 66
- [MS15] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2015. 66
- [MSD23] Kirill Mazur, Edgar Sucar, and Andrew J Davison. Feature-realistic neural fusion for real-time, open set scene understanding. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 19
- [MSGH16] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 58
- [MSOC⁺19] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019. 104
- [MST⁺20a] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 5, 15, 16, 66, 77, 79, 95

- [MST⁺20b] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 40
- [MWG⁺21] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021. 59
- [NBM⁺22] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 67
- [NG21] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 46
- [NIH⁺11] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011. 4, 18
- [NJD⁺22] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 68, 78, 81
- [NLD11] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011. 18

- [NMOG20] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 66
- [NMS18] Lachlan Nicholson, Michael Milford, and Niko Sünderhauf. Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam. *IEEE Robotics and Automation Letters*, 2018. 5, 19
- [OPG21] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 44
- [PEL⁺24] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 83
- [PFS⁺19] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4, 15, 40, 66
- [PHBB⁺24] Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holzheimer, Christos Kaplanis, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, Stephen Spencer, Jessica Yung, Michael Dennis, Sultan Kenjeyev, Shangbang Long, Vlad Mnih, Harris Chan, Maxime Gazeau, Bonnie Li, Fabio Pardo, Luyu Wang, Lei Zhang, Frederic Besse, Tim Harley, Anna Mitenkova, Jane Wang, Jeff Clune, Demis Hassabis, Raia Hadsell, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 2: A large-scale foundation world model. <https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/>, 2024. 8, 35, 93, 96

- [PJBM23] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. 67, 79, 95
- [PNM⁺20] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 66
- [PQFS24] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 71
- [PSDV⁺18] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2018. 25
- [QSMG17] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 66
- [QYSG17] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 66
- [RA17a] Martin Rünz and Lourdes Agapito. Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 19
- [RA17b] Martin Rünz and Lourdes Agapito. Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 40
- [RBA18] Martin Runz, Maud Buffier, and Lourdes Agapito. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2018. 19

- [RBL⁺22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 7, 27, 67, 70, 97, 99, 102
- [RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2015. 25, 27
- [RGA⁺20] Antoni Rosinol, Arjun Gupta, Marcus Abate, Jingnan Shi, and Luca Carlone. 3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans. In *Proceedings of Robotics: Science and Systems (RSS)*, 2020. 16
- [RHB⁺25] Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. Gaia-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025. 93
- [RJY⁺21] Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. Derf: Decomposed radiance fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 40
- [RKH⁺21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. 71
- [RLC23] Antoni Rosinol, John J Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2023. 40
- [RNS⁺18] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI blog*, 2018. 32

- [Ros22] Toni Rosinol. History of visual slam [tweet]. <https://x.com/RosinolToni/status/1590752097339047938>, 2022. 17
- [RPLG21] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 40, 43
- [RSR⁺20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, 2020. 30
- [Rud17] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 58
- [SCL⁺25] Wenqiang Sun, Shuo Chen, Fangfu Liu, Zilong Chen, Yueqi Duan, Jun Zhang, and Yikai Wang. Dimensionx: Create any 3d and 4d scenes from a single image with controllable video diffusion. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2025. 95
- [SCP⁺23] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 113
- [SCZ⁺23] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023. 65, 68, 79
- [SDCS23] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2023. 26, 109
- [SDWMG15] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015. 7, 23

- [SEE⁺12] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A Benchmark for the Evaluation of RGB-D SLAM Systems. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2012. 53
- [SLBR24] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 109
- [SLOD21] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 19, 39, 40, 47, 49, 53, 59
- [SLP⁺21] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. 72, 74, 100, 114
- [SLS⁺24] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, et al. Zeronvs: Zero-shot 360-degree view synthesis from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 99
- [SME21] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 24, 109
- [SMK25] Katja Schwarz, Norman Mueller, and Peter Kotschieder. Generative gaussian splatting: Generating 3d scenes with video diffusion priors. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2025. 113
- [SMNS⁺13] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 5, 19, 39

- [SSC22] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5, 64
- [SSDK⁺20] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 7, 23, 25
- [SWD20] Edgar Sucar, Kentaro Wada, and Andrew Davison. NodeSLAM: Neural object descriptors for multi-view shape reconstruction. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2020. 5, 16, 19, 40
- [SWM⁺19] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 46
- [SWY⁺24] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 28, 68, 83
- [SZS⁺25] Stanislaw Szymanowicz, Jason Y. Zhang, Pratul Srinivasan, Ruiqi Gao, Arthur Brussee, Aleksander Holynski, Ricardo Martin-Brualla, Jonathan T. Barron, and Philipp Henzler. Bolt3D: Generating 3D Scenes in Seconds. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2025. 113
- [SZW19] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 66, 94, 96, 100
- [TDBM22] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 2022. 115
- [Tea24] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024. 109

- [TET⁺22] Towaki Takikawa, Alex Evans, Jonathan Tremblay, Thomas Müller, Morgan McGuire, Alec Jacobson, and Sanja Fidler. Variable bitrate neural fields. In *Proceedings of SIGGRAPH*, 2022. 66
- [TJS⁺25] Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, WQ Zhang, Weifeng Luo, et al. Magi-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025. 95
- [TJY⁺24] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 32
- [TLK⁺23] Hung-Yu Tseng, Qinbo Li, Changil Kim, Suhub Alsisan, Jia-Bin Huang, and Johannes Kopf. Consistent view synthesis with pose-guided diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 95
- [TME⁺22] Jonathan Tremblay, Moustafa Meshry, Alex Evans, Jan Kautz, Alexander Keller, Sameh Khamis, Thomas Müller, Charles Loop, Nathan Morrical, Koki Nagano, et al. Rtmv: A ray-traced multi-view synthetic dataset for novel view synthesis. *arXiv preprint arXiv:2205.07058*, 2022. 78
- [TRZ⁺24] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 95
- [VdOKE⁺16] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 32
- [VDOKK16] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2016. 32

- [VGVG⁺21] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2021. 58
- [VLAF25] Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. 8, 35, 93, 94, 96
- [VSP⁺17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 21, 22, 31, 65
- [VYB⁺24] Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 89
- [WA24] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2024. 20, 96
- [WBA22] Jingwen Wang, Tymoteusz Bleja, and Lourdes Agapito. Go-surf: Neural feature grid optimization for fast, high-fidelity rgb-d surface reconstruction. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2022. 40
- [WBSS04] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004. 77
- [WCK⁺25] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 20, 114

- [WDH⁺23] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 71
- [WGP⁺25] Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T. Barron, and Aleksander Holynski. CAT4D: Create Anything in 4D with Multi-View Video Diffusion Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 89, 95, 109, 114
- [WLC⁺22] Qianyi Wu, Xian Liu, Yuedong Chen, Kejie Li, Chuanxia Zheng, Jianfei Cai, and Jianmin Zheng. Object-compositional neural implicit surfaces. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 40, 50
- [WLC⁺24] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 19, 20, 95, 104, 114
- [WLL⁺21] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 81, 83
- [WLL⁺22] Philippe Weinzaepfel, Vincent Leroy, Thomas Lucas, Romain Brégier, Yohann Cabon, Vaibhav Arora, Leonid Antsfeld, Boris Chidlovskii, Gabriela Csurka, and Jérôme Revaud. Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 114
- [WLL⁺23] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. Croco v2: Improved cross-view completion pre-training for stereo matching and optical flow. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 114

- [WLSM⁺15] Thomas Whelan, Stefan Leutenegger, Renato F Salas-Moreno, Ben Glocker, and Andrew J Davison. Elasticfusion: Dense slam without a pose graph. In *Proceedings of Robotics: Science and Systems (RSS)*, 2015. 18
- [WMK⁺25] Ethan Weber, Norman Müller, Yash Kant, Vasu Agrawal, Michael Zollhöfer, Angjoo Kanazawa, and Christian Richardt. Fillerbuster: Multi-view scene completion for casual captures. *arXiv preprint arXiv:2502.05175*, 2025. 114
- [WRA21] Jingwen Wang, Martin Rünz, and Lourdes Agapito. Dsp-slam: object oriented slam with deep shape priors. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2021. 40
- [WSJ⁺20] Kentaro Wada, Edgar Sucar, Stephen James, Daniel Lenton, and Andrew J Davison. Morefusion: Multi-object reasoning for 6d pose estimation from volumetric fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 60
- [WSK⁺15] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 66
- [WSL⁺24] Daniel Watson, Saurabh Saxena, Lala Li, Andrea Tagliasacchi, and David J Fleet. Controlling space and time with diffusion models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 93, 94, 95, 99, 100, 102, 104, 105, 106, 107, 109
- [WWA23] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 19
- [WYW⁺24] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *Proceedings of SIGGRAPH*, 2024. 95, 96, 102, 104, 105

- [WZ89] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1989. 99
- [WZH⁺25] Qianqian Wang*, Yifei Zhang*, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 20, 96, 102
- [XJW⁺23] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Yi Wang, and Zhangyang Wang. Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360deg views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 68
- [XLH⁺25] Jing Xiong, Gongye Liu, Lun Huang, Chengyue Wu, Taiqiang Wu, Yao Mu, Yuan Yao, Hui Shen, Zhongwei Wan, Jinfa Huang, et al. Autoregressive models in vision: A survey. *Transactions on Machine Learning Research (TMLR)*, 2025. 30, 31
- [XLT⁺19] Binbin Xu, Wenbin Li, Dimos Tzoumanikas, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Mid-fusion: Octree-based object-level multi-instance dynamic slam. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2019. 5, 16, 19, 40
- [XLZ⁺25] Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. Worldmem: Long-term consistent world simulation with memory. *arXiv preprint arXiv:2504.12369*, 2025. 96, 102
- [XMB⁺25] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. 109
- [XPW⁺25] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2025. 95, 102, 104, 105, 113

- [XXK⁺24] Sirui Xie, Zhisheng Xiao, Diederik Kingma, Tingbo Hou, Ying Nian Wu, Kevin P Murphy, Tim Salimans, Ben Poole, and Ruiqi Gao. Em distillation for one-step diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 109
- [XXP⁺22] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5
- [YAAF⁺24] Jason J. Yu, Tristan Aumentado-Armstrong, Fereshteh Forghani, Konstantinos G. Derpanis, and Marcus A. Brubaker. Polyoculus: Simultaneous multi-view image-based novel view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 93, 95
- [Yan22] Lecun Yann. From machine learning to autonomous intelligence. <https://leshouches2022.github.io/SLIDES/compressed-yann-1.pdf>, 2022. 34
- [YCL⁺25] Yijun Yuan, Zhuoguang Chen, Kenan Li, Weibang Wang, and Hang Zhao. Slam-former: Putting slam into one transformer. *arXiv preprint arXiv:2509.16909*, 2025. 20
- [YCS⁺23] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 32
- [YFDB23] Jason J. Yu, Fereshteh Forghani, Konstantinos G. Derpanis, and Marcus A. Brubaker. Long-term photometric consistent novel view synthesis with diffusion models. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 95, 102, 104, 105, 107

- [YLG⁺24] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion—tokenizer is key to visual generation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 32
- [YLY⁺18] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 64
- [YLZ⁺22] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-Fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2022. 40
- [YPN⁺22] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 60
- [YPW23] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 67
- [YQX⁺24] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 19
- [YS19] Shichao Yang and Sebastian Scherer. Cubeslam: Monocular 3-d object slam. *IEEE Transactions on Robotics (T-RO)*, 2019. 19
- [YSL⁺22] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 46

- [YWL⁺24] Jianglong Ye, Peng Wang, Kejie Li, Yichun Shi, and Heng Wang. Consistent-1-to-3: Consistent image to 3d view synthesis via geometry-aware diffusion models. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2024. 68, 79
- [YXY⁺25] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2025. 95, 102, 104, 105, 106
- [YYTK21] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 67
- [YZAS21] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021. 32
- [YZX⁺21] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 40, 44
- [YZZ⁺25] Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 95, 109
- [ZBLD19] Shuaifeng Zhi, Michael Bloesch, Stefan Leutenegger, and Andrew J Davison. SceneCode: Monocular dense semantic reconstruction using learned encoded scene representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 40
- [ZGD⁺20] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al.

- Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 30
- [ZGJ⁺22] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 47, 52
- [ZGV⁺25] Jensen (Jinghao) Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2025. 89, 93, 94, 95, 99, 102, 104, 105, 106, 107, 108, 109
- [ZIE⁺18] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 77
- [ZK13] Qian-Yi Zhou and Vladlen Koltun. Dense scene reconstruction with points of interest. *ACM Transactions on Graphics (TOG)*, 2013. 39, 49
- [ZPBS23] Xingguang Zhong, Yue Pan, Jens Behley, and Cyrill Stachniss. Shine-mapping: Large-scale 3d mapping using sparse hierarchical implicit neural representations. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 40
- [ZPL⁺22] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 19, 40, 47, 49, 53
- [ZSM⁺22] Shuaifeng Zhi, Edgar Sucar, Andre Mouton, Iain Haughton, Tristan Laidlow, and Andrew J Davison. ilabel: Revealing objects in neural fields. *IEEE Robotics and Automation Letters*, 2022. 40

- [ZTF⁺18] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *Proceedings of SIGGRAPH*, 2018. 104, 107
- [ZYB⁺25] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. 109
- [ZYG⁺24] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 113