

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

From 3D Reconstruction to Spatial Intelligence: Bridging Vision Foundation Models with the 3D World

Permalink

<https://escholarship.org/uc/item/3zd8s0dm>

ISBN

9798193220794

Author

Fu, Yang

Publication Date

2026-04-09

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

From 3D Reconstruction to Spatial Intelligence:
Bridging Vision Foundation Models with the 3D World

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Electrical Engineering (Signal and Image Processing)

by

Yang Fu

Committee in charge:

Professor Xiaolong Wang, Chair
Professor Zhuowen Tu
Professor Nuno M Vasconcelos
Professor Rose Yu

Copyright

Yang Fu, 2026

All rights reserved.

The Dissertation of Yang Fu is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2026

TABLE OF CONTENTS

Dissertation Approval Page	iii
Table of Contents	iv
List of Figures	vii
List of Tables	viii
Acknowledgements	x
Vita	xii
Abstract of the Dissertation	xiii
Introduction	1
Chapter 1 Scalable 3D Data Curation from Pose-Free Reconstruction	4
1.1 Introduction	4
1.2 Preliminary: 3D Gaussian Splatting	6
1.3 COLMAP-Free 3D Gaussian Splatting	8
1.3.1 Local 3DGS for Relative Pose Estimation	8
1.3.2 Global 3DGS with Progressively Growing	11
1.4 Experiments	12
1.4.1 Experimental Setup	12
1.4.2 Comparing with Pose-Free Methods	13
1.4.3 Results on Scenes with Large Camera Motions	15
1.4.4 Ablation Study	16
1.5 Conclusion	18
Chapter 2 Scalable 3D Data Curation from Generative Models	20
2.1 Introduction	20
2.2 Preliminary: Diffusion Models	22
2.3 Hand-Object-Interaction Diffusion Model	23
2.3.1 Hand Grasping Trajectory Generation	24
2.3.2 Hand-Object-Interaction image Synthesis	25
2.4 Experiment	27
2.4.1 Implementation Details	27
2.4.2 Evaluation and Baseline Comparison	27
2.4.3 Geometry and Appearance Disentangle	30
2.4.4 Ablation Study	31
2.4.5 Applications	32
2.5 Conclusion	35

Chapter 3	Bridging 2D Vision Language Models to the 3D World	36
3.1	Introduction	36
3.2	Method	39
3.2.1	3D Gaussian Reconstruction	39
3.2.2	Multi-view Feature Fusion	40
3.2.3	3D Feature Distillation	41
3.2.4	Inference	43
3.3	Experiments	44
3.3.1	Experimental Setup	44
3.3.2	Evaluation on Scene Understanding	45
3.3.3	Evaluation on 3D Question Answering	47
3.3.4	Evaluation on 3D Dense Captioning	48
3.3.5	Evaluation on 3D Visual Grounding	49
3.3.6	Ablation Study	50
3.4	Conclusion	52
Chapter 4	3D Perception in the Wild — Category-Level 6D Object Pose Estimation .	54
4.1	Introduction	54
4.2	Problem Setting	57
4.3	Wild6D Dataset	58
4.4	Method	59
4.4.1	Framework Details	60
4.4.2	Learning Objectives	64
4.5	Experiments	66
4.5.1	Implementation Details	66
4.5.2	Ablation Study	66
4.5.3	Comparison with State-of-the-art Methods	68
4.6	Conclusion	70
Chapter 5	3D Perception in the Wild — Scene Reconstruction as Mapping Priors for Autonomous Driving	71
5.1	Introduction	71
5.2	Method	73
5.2.1	Scalable Mapping Prior Generation	73
5.2.2	Mapping Prior Augmented 3D Detection	75
5.2.3	Mixed-Modality Training Strategy	77
5.2.4	Training Objectives	78
5.3	Experiments	79
5.3.1	Datasets and Metrics	79
5.3.2	Implementation Details	80
5.3.3	Comparison with state-of-the-art works	81
5.3.4	Ablation study	83
5.4	Conclusion	85

Chapter 6	Conclusion and Future Work	86
Bibliography		90

LIST OF FIGURES

Figure 1.1.	Novel View Synthesis and Camera Pose Estimation Comparison	5
Figure 1.2.	Overview of proposed CF-3DGS	9
Figure 1.3.	Qualitative comparison on Tanks and Temples	14
Figure 1.4.	Qualitative Comparison on CO3D V2	16
Figure 2.1.	HOIDiffusion Overview	21
Figure 2.2.	HOIDiffusion Inference Pipeline	24
Figure 2.3.	HOIDiffusion Architecture	26
Figure 2.4.	Qualitative results on different structures conditions	29
Figure 2.5.	Qualitative Results of HOIDiffusion with different background prompts ..	31
Figure 2.6.	Ablation study on appearance regularization	33
Figure 2.7.	Zero-shot hand grasping trajectory generation	34
Figure 3.1.	SplatDistill Overview	37
Figure 3.2.	SplatDistill Architecture	41
Figure 3.3.	Visualization of Feature Distillation	47
Figure 4.1.	Wild6D Dataset	55
Figure 4.2.	RepoNet Semi-supervised setting	60
Figure 4.3.	RepoNet Framework	63
Figure 5.1.	Illustration of Scene Reconstruction as Mapping Priors for 3D Detection .	72
Figure 5.2.	Architecture of MPA3D	74

LIST OF TABLES

Table 1.1.	Novel view synthesis results on Tanks and Temples	13
Table 1.2.	Pose accuracy on Tanks and Temples	15
Table 1.3.	Novel view synthesis results on CO3D V2	15
Table 1.4.	Pose accuracy on CO3D V2	16
Table 1.5.	Progressively Growing on Tanks and Temple	17
Table 1.6.	Ablation study of depth loss on Tanks and Temples.	17
Table 1.7.	Comparison with 3DGS reconstructed from COLMAP poses	18
Table 2.1.	Quantitative results of Appearance Alignment on DexYCB	28
Table 2.2.	Hand Contact Recall Evaluation	29
Table 2.3.	Ablation study on different geometric structural maps	32
Table 2.4.	Ablation study on appearance regularization	32
Table 2.5.	Quantitative results of HOIDiffusion on NOCS	34
Table 3.1.	3D object recognition result on ScanNetv2	45
Table 3.2.	3D instance segmentation results on ScanNet200	46
Table 3.3.	Quantitative result on ScanQA	48
Table 3.4.	Quantitative results on Scan2Cap	49
Table 3.5.	Quantitative result on ScanRefer	50
Table 3.6.	Ablation on Rendering loss	50
Table 3.7.	Efficiency Analysis of SplatDistill	51
Table 3.8.	Comparision with existing 3D VLMs	51
Table 3.9.	Ablation on the number of sampling views	52
Table 4.1.	Comparison between Wild6D and existing datasets.	58
Table 4.2.	Ablation on Implicit function and intermediate NOCS map	67

Table 4.3.	Ablation study on pose regression loss and silhouette matching loss	67
Table 4.4.	Ablation study on different training data	68
Table 4.5.	Comparison with the SOTAs on REAL275	69
Table 4.6.	Comparison with the SOTAs on Wild6D	69
Table 5.1.	Performance comparisons on WOD validation set	81
Table 5.2.	Comparisons with temporal fusion based approaches	82
Table 5.3.	Results on WOD test set leaderboard	82
Table 5.4.	Ablation study on different Mapping Priors	83
Table 5.5.	Ablation study on different fusion strategies	84

ACKNOWLEDGEMENTS

This thesis would not have been possible without the support, guidance, and companionship of many wonderful people.

First and foremost, I am deeply grateful to my advisor, Xiaolong Wang, for giving me the freedom to explore and pursue research directions I am truly passionate about. His exceptional taste in research and unwavering support throughout my PhD have shaped not only this thesis, but the way I think about science. I could not have asked for a better mentor to grow under.

I am also sincerely thankful to Sifei Liu at NVIDIA, whose hands-on guidance and patient mentorship made a tremendous difference during critical stages of my research. His willingness to dive deep into problems alongside me and offer thoughtful advice was invaluable.

I owe a great deal to Anjie, an outstanding collaborator whose deep technical insight and openness to discussion and brainstorming have enriched nearly every project we worked on together. Many of the ideas in this thesis were sharpened through our conversations. I am equally grateful to my labmates and friends, Jiarui Xu and Jiteng Mu, for the camaraderie, the lively discussions, and for making the long days of research genuinely enjoyable.

Finally, and most deeply, I thank my parents for their unconditional love and steadfast support throughout this journey. They have always believed in me, even when the path was uncertain, and their encouragement has been my greatest source of strength.

Chapter 1, in full, is based on the paper from Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros and Xiaolong Wang, “COLMAP-Free 3D Gaussian Splatting” published in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. The dissertation author is the primary investigator and author of this paper.

Chapter 2, in full, is based on the paper from Mengqi Zhang, Yang Fu, Zheng Ding, Sifei Liu, Zhuowen Tu, and Xiaolong Wang, “HOIDiffusion: Generating Realistic 3D Hand-Object Interaction Data” published in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024. The dissertation author is the primary investigator and author of this paper.

Chapter 3, in part, is based on the paper from Yang Fu, Sifei Liu, Hongxu Yin, Benjamin Eckart, Jan Kautz, Xiaolong Wang, Arash Vahdat and Chao Liu, “Bridging 2D Vision Language Models to 3D World via Feature Distillation” currently being prepared for submission. The dissertation author is the primary investigator and author of this paper.

Chapter 4, in part, is based on the paper from Yang Fu and Xiaolong Wang, “Category-Level 6D Object Pose Estimation in the Wild: A Semi-Supervised Learning Approach and A New Dataset” published in Advances in Neural Information Processing Systems (NeurIPS), 2022. The dissertation author is the primary investigator and author of this paper.

Chapter 5, in part, is based on the paper from Yang Fu, Yuliang Zou, Hao Xiang, Xin Huang, Yijing Bai, Chen Song, Weijing Shi, Govind Thattai, Dragomir Anguelov, Mingxing Tan, and Yingwei Li, “Scene Reconstruction as Mapping Priors for 3D Detection” has been accepted and will be published in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2026. The dissertation author is the primary investigator and author of this paper.

VITA

- | | |
|------|--|
| 2017 | Bachelor of Science in Electronic Information Engineering, Beihang University |
| 2020 | Master of Science in Electrical and Computer Engineering, University of Illinois Urbana-Champaign |
| 2026 | Doctor of Philosophy in Electrical Engineering (Signal and Image Processing), University of California San Diego |

ABSTRACT OF THE DISSERTATION

From 3D Reconstruction to Spatial Intelligence:
Bridging Vision Foundation Models with the 3D World

by

Yang Fu

Doctor of Philosophy in Electrical Engineering (Signal and Image Processing)

University of California San Diego, 2026

Professor Xiaolong Wang, Chair

Understanding the 3D world from visual observations is a fundamental challenge in computer vision, with broad implications for robotics, autonomous driving, and embodied AI. Despite remarkable progress in 2D vision, a critical disconnect persists: existing foundation models are anchored to the 2D image plane and lack the geometric grounding needed to reason about 3D environments. At the root of this disconnect lies a data problem — 3D data is expensive, difficult to scale, and insufficient to support the kind of large-scale training that has driven 2D progress. This thesis addresses these challenges across three interconnected thrusts: scalable 3D data curation, spatial reasoning, and real-world 3D perception.

First, I tackle the 3D data bottleneck from two complementary perspectives. From a reconstruction standpoint, I present COLMAP-Free 3D Gaussian Splatting, which eliminates the dependence on structure-from-motion preprocessing, enabling scalable 3D reconstruction from casually captured video. From a generative standpoint, I present HOIDifusion, which leverages diffusion models to generate realistic 3D hand-object interaction data, addressing a data gap that is particularly critical for embodied AI. Second, I investigate how to bridge the gap between 2D foundation models and the 3D world. I propose a feature distillation framework that transfers 3D structural knowledge into 2D vision-language models, equipping them with spatial reasoning capabilities without requiring full 3D supervision at inference time. Third, I push 3D perception toward robust real-world deployment. I address category-level 6D object pose estimation in the wild through semi-supervised learning, and demonstrate how scene reconstruction can serve as structural mapping priors to improve 3D object detection for autonomous driving.

Together, these contributions form a cohesive path from raw visual data to semantically rich, spatially aware intelligence — advancing the long-term goal of machines that can perceive, reason about, and act within the 3D world.

Introduction

The past decade has witnessed transformative progress in computer vision, driven largely by the success of deep learning on large-scale 2D image and video data. Foundation models trained on billions of images have demonstrated remarkable capabilities in recognition, segmentation, and language-grounded visual understanding. Yet despite this progress, a fundamental limitation persists: these models are inherently anchored to the 2D image plane. They perceive the world as a collection of pixels, without an intrinsic understanding of the three-dimensional structure that underlies every visual observation. For machines to truly perceive and reason about the world — as humans do — bridging this gap between 2D foundation models and the 3D world is essential.

This disconnect manifests in many practical failure modes. A state-of-the-art vision-language model can identify that a cup is "to the left of the keyboard" in an image, but cannot reliably estimate how far away it is, whether it is in front of or behind another object in 3D space, or how its pose would change from a different viewpoint. Similarly, autonomous driving systems that rely primarily on 2D perception struggle to accurately localize objects in 3D, reason about occlusion, or generalize to novel environments. As AI systems are increasingly deployed in embodied and interactive settings — robotics, autonomous vehicles, augmented reality — the inability to ground perception in three dimensions becomes a critical bottleneck.

The remarkable success of 2D foundation models rests on one key ingredient: data. Models like CLIP, DINO, and GPT-4V benefit from internet-scale supervision — billions of image-text pairs that provide rich, diverse, and naturally annotated training signal. The 3D world, however, lacks an equivalent. Acquiring accurate 3D data is expensive, labor-intensive, and

difficult to scale. Traditional pipelines rely on controlled capture setups, depth sensors, or manual annotation, all of which impose significant constraints on diversity and scale. As a result, 3D perception and reconstruction models have historically been trained on small, domain-specific datasets that fail to capture the complexity and variability of real-world environments.

This data scarcity creates a vicious cycle: without large-scale 3D data, models cannot learn robust 3D representations; without robust representations, it is difficult to build the automated pipelines needed to generate more 3D data. Breaking this cycle requires rethinking how we acquire, represent, and leverage 3D information — starting from raw, unconstrained visual observations and building upward toward rich semantic understanding.

Thesis Overview and Contributions

This thesis addresses the disconnect between 2D foundation models and the 3D world through three interconnected research thrusts — data curation, spatial reasoning, and real-world perception — organized across five technical chapters.

Chapter 1: Scalable 3D Data Curation from Pose-Free Reconstruction. The first thrust tackles the 3D data bottleneck from a reconstruction perspective. Traditional 3D reconstruction pipelines depend heavily on structure-from-motion preprocessing, which is computationally expensive, prone to failure in textureless or dynamic scenes, and fundamentally limits scalability. I present COLMAP-Free 3D Gaussian Splatting, a method that eliminates this dependency by jointly optimizing camera poses and 3D Gaussian representations directly from image sequences. This enables scalable, high-quality 3D reconstruction from casually captured video without any preprocessing, opening the door to large-scale 3D data acquisition in the wild.

Chapter 2: Scalable 3D Data Curation from Generative Models. While reconstruction approaches capture real-world scenes, they remain limited by what can be physically observed and captured. The second data-curation thrust takes a complementary approach: generating realistic 3D training data through generative modeling. I present HOIDifusion, a framework for generating realistic 3D hand-object interaction data by leveraging diffusion models conditioned

on 3D geometry. This addresses a particularly challenging data gap — hand-object interactions are difficult to capture at scale but critical for robotics and embodied AI applications.

Chapter 3: Bridging 2D Vision Language Models to the 3D World. With scalable 3D data in hand, the second research thrust investigates how to close the gap between powerful 2D foundation models and 3D understanding. I present a feature distillation framework that transfers 3D structural knowledge into 2D vision-language models, equipping them with spatial awareness without requiring full 3D supervision at inference time. This work demonstrates that the representational richness of 3D data can be effectively distilled into the scalable architecture of 2D VLMs, enabling genuine spatial reasoning capabilities.

Chapter 4: 3D Perception in the Wild — Category-Level 6D Object Pose Estimation. The third thrust pushes toward robust 3D perception in unconstrained real-world settings. I address category-level 6D object pose estimation, where the goal is to estimate the full 3D position and orientation of objects across diverse categories without instance-specific models. I develop approaches based on semi-supervised learning and self-supervised geometric correspondence that generalize across object categories and challenging capture conditions, significantly advancing the state of the art on in-the-wild pose estimation.

Chapter 5: 3D Perception in the Wild — Scene Reconstruction as Mapping Priors for Autonomous Driving. The final technical chapter scales 3D perception to the autonomous driving setting, where accuracy, generalization, and efficiency are all critical. I present a framework that leverages scene reconstruction as structural mapping priors for 3D object detection, enabling detection models to exploit rich geometric context beyond what is available from raw sensor inputs alone. This work demonstrates that reconstruction and perception, often treated as separate problems, can be deeply integrated to mutual benefit.

Finally, **Chapter 6** concludes the thesis with a discussion of limitations and promising directions for future research.

Chapter 1

Scalable 3D Data Curation from Pose-Free Reconstruction

1.1 Introduction

The field of photo-realistic scene reconstruction and view synthesis has been largely advanced with the rise of Neural Radiance Fields (NeRFs [97]). An important initialization step for training NeRF is to first prepare the camera poses for each input image. This is usually achieved by running the Structure-from-Motion (SfM) library COLMAP [129]. However, this pre-processing is not only time-consuming but also can fail due to its sensitivity to feature extraction errors and difficulties in handling textureless or repetitive regions.

Recent studies [157, 80, 9] have focused on reducing the reliance on SfM by integrating pose estimation directly within the NeRF framework. Simultaneously solving 3D scene reconstruction and camera registration has been a chicken-and-egg problem for a long time in computer vision. This challenge is further amplified in the context of NeRF and its implicit representation, where the optimization process often involves additional constraints. For instance, BARF [80] requires initial poses that are close to their ground truth locations, and NeRFmm [157] is largely limited to face-forwarding scenes. The recently proposed Nope-NeRF [9] takes a long time to train (30 hours) and does not work well when the camera pose changes a lot (e.g., 360 degrees), as shown in the two top cases in Fig. 1.1. Fundamentally, NeRFs optimize camera parameters in an indirect way, by updating the ray casting from camera positions, which makes optimization

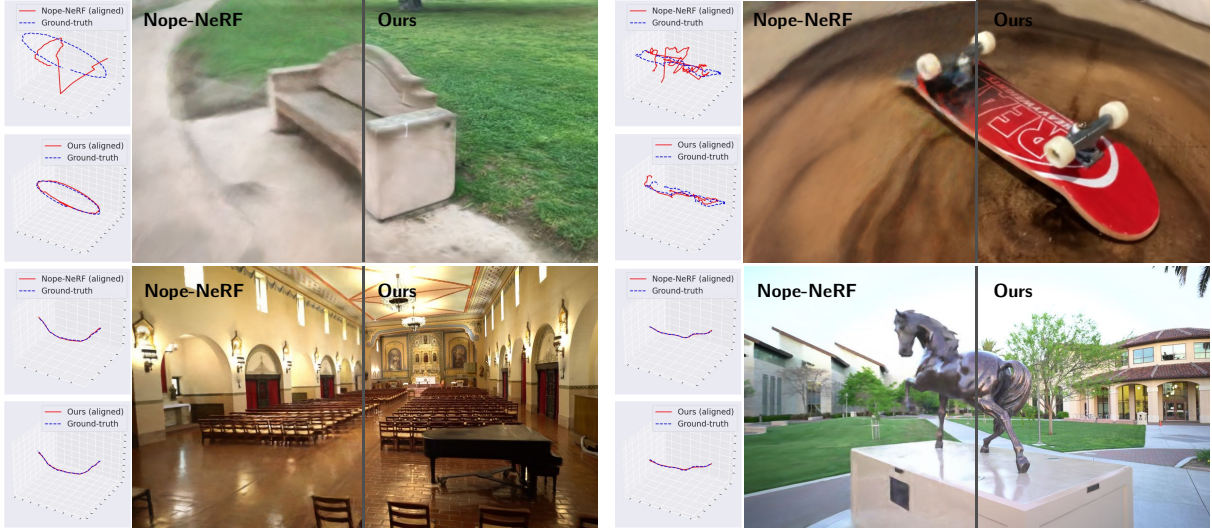


Figure 1.1. Novel View Synthesis and Camera Pose Estimation Comparison. We propose COLMAP-Free 3D Gaussian Splatting (CF-3DGS) for novel view synthesis without known camera parameters. Our method achieves more robustness in pose estimation and better quality in novel view synthesis than previous state-of-the-art methods.

challenging.

The arrival of 3D Gaussian Splatting [60] extends the volumetric rendering in NeRFs to accommodate point clouds. While it was originally proposed with pre-computed cameras, we find it offers a new opportunity to perform view synthesis without SfM pre-processing. We propose **COLMAP-Free 3D Gaussian Splatting (CF-3DGS)**, which leverages two key ingredients: the **temporal continuity from video** and the **explicit point cloud representation**. We summarize our approach below.

Instead of optimizing with all the frames at once, we propose to build the 3D Gaussians of the scene in a continuous manner, “growing” one frame at a time as the camera moves. In this process, we will extract a *local* 3D Gaussians set for each frame, and also maintain a *global* 3D Gaussians set of the whole scene. Assuming we are iterating through $t = \{1, \dots, T\}$ frames in a sequential manner, we perform a two-step procedure each time: (i) We construct a local 3D Gaussians set given frame $t - 1$, and we sample the next nearby frame t . Our goal is to learn an affine transformation that can transform the 3D Gaussians in frame $t - 1$ to render the pixels in

frame t . Neural rendering provides the gradients for optimizing the affine transformation, which is essentially the relative camera pose between frames $t - 1$ and t . This optimization is not difficult as the **explicit** point cloud representation allows us to directly apply an affine transformation on it which cannot be achieved with NeRFs, and the two frames are close (**temporal continuity**) which makes the transformation relatively small. (ii) Once we have the relative camera pose between frames $t - 1$ and t , we can infer the relative pose between the first frame and frame t . This allows us to aggregate the current frame information into the global 3D Gaussians set, where we will perform optimization with the current and all the previous frames and camera poses.

We experiment with two datasets: the Tanks and Temples dataset [68] and videos randomly selected from the CO3D dataset [123]. We evaluate both view synthesis and camera pose estimation tasks, and compare with previous approaches without pre-computed camera poses. Our method performs significantly better than previous approaches on view synthesis in both datasets. For camera pose estimation, we find our method performs on par with the most recent Nope-NeRF [9] when the camera motion is small, and outperforms all approaches by a large margin when the camera changes a lot, such as in the 360-degree videos in CO3D.

1.2 Preliminary: 3D Gaussian Splatting

3DGS [60] models the scene as a set of 3D Gaussians, which is an explicit form of representation, in contrast to the implicit representation used in NeRFs. Each Gaussian is characterized by a covariance matrix Σ and a center (mean) point μ ,

$$G(x) = e^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)} \quad (1.1)$$

The means of 3D Gaussians are initialized by a set of sparse point clouds(*e.g.*, always obtained from SfM). Each Gaussian is parameterized as the following parameters: (a) a center position $\mu \in \mathbb{R}^3$; (b) spherical harmonics (SH) coefficients $c \in \mathbb{R}^k$ (k represents the degrees of freedom)

that represents the color; (c) rotation factor $r \in \mathbb{R}^4$ (in quaternion rotation); (d) scale factor $s \in \mathbb{R}^3$; (e) opacity $\alpha \in \mathbb{R}$. Then, the covariance matrix Σ describes an ellipsoid configured by a scaling matrix $S = \text{diag}([s_x, s_y, s_z])$ and rotation matrix $R = q2R([r_w, r_x, r_y, r_z])$, where $q2R()$ is the formula for constructing a rotation matrix from a quaternion. Then, the covariance matrix can be computed as follows,

$$\Sigma = RSS^\top R^\top \quad (1.2)$$

In order to optimize the parameters of 3D Gaussians to represent the scene, we need to render them into images in a differentiable manner. As introduced in [60], the rendering from a given camera view W involves the process of splatting the Gaussian onto the image plane, which is achieved by approximating the projection of a 3D Gaussian along the depth dimension into pixel coordinates. Given a viewing transform W (also known as the camera pose), the covariance matrix Σ^{2D} in camera coordinates can be expressed by

$$\Sigma^{2D} = JW\Sigma W^\top J^\top \quad (1.3)$$

where J is the Jacobian of the affine approximation of the projective transformation. For each pixel, the color and opacity of all the Gaussians are computed using Eq. 1.1, and the final rendered color can be formulated as the alpha-blending of N ordered points that overlap the pixel,

$$C_{pix} = \sum_i^N c_i \alpha_i \prod_j^{i-1} (1 - \alpha_j) \quad (1.4)$$

where c_i, α_i represents the density and color of this point computed from the learnable per-point opacity and SH color coefficients weighted by the Gaussian covariance Σ , which we ignore in Eq. 1.4 for simplicity.

To perform scene reconstruction, given the ground truth poses that determine the projections, we fit a set of initialized Gaussian points to the desired objects or scenes by learning their parameters, i.e., μ and Σ . With the differentiable renderer as in Eq. 1.4, all those parameters,

along with the SH and opacity, can be easily optimized through a photometric loss. In our approach, we reconstruct scenes following the same process, but replacing the ground truth poses with the estimated ones, as detailed in the following sections.

1.3 COLMAP-Free 3D Gaussian Splatting

Given a sequence of unposed images along with camera intrinsics, our goal is to recover the camera poses and reconstruct the photo-realistic scene. To this end, we propose CF-3DGS to optimize the 3D Gaussian Splatting (3DGS [60]) and camera poses simultaneously. We detail our method in the following sections: first, we propose a *local* 3DGS, a simple yet effective method to estimate the relative camera pose from each pair of nearby frames, in Sec. 1.3.1. Then, we introduce a *global* 3DGS, featuring a progressive expansion of the 3D Gaussians from unobserved views in sequential order, in Sec. 1.3.2.

1.3.1 Local 3DGS for Relative Pose Estimation

Previous studies [80, 59, 9] have demonstrated the feasibility of simultaneously estimating camera parameters and optimizing a Neural Radiance Field (NeRF). This typically involves the integration of various regularization terms and geometric priors. However, rather than directly optimizing camera poses, most existing methods prioritize optimizing the ray casting process from varying camera positions. This is dictated by the nature of implicit representation and the implementation of ray tracing in NeRFs. This indirect approach often results in a complex and challenging optimization under large camera movement scenarios.

On the other hand, 3DGS [60] utilizes an explicit scene representation in the form of point clouds enabling straightforward deformation and movement, as demonstrated in its recent application to dynamic scenes [92, 160]. To take advantage of 3DGS, we introduce a *local* 3DGS to estimate the relative camera pose.

We reveal the relationship between the camera pose and the 3D rigid transformation of Gaussian points, in the following. Given a set of 3D Gaussians with centers μ , projecting them

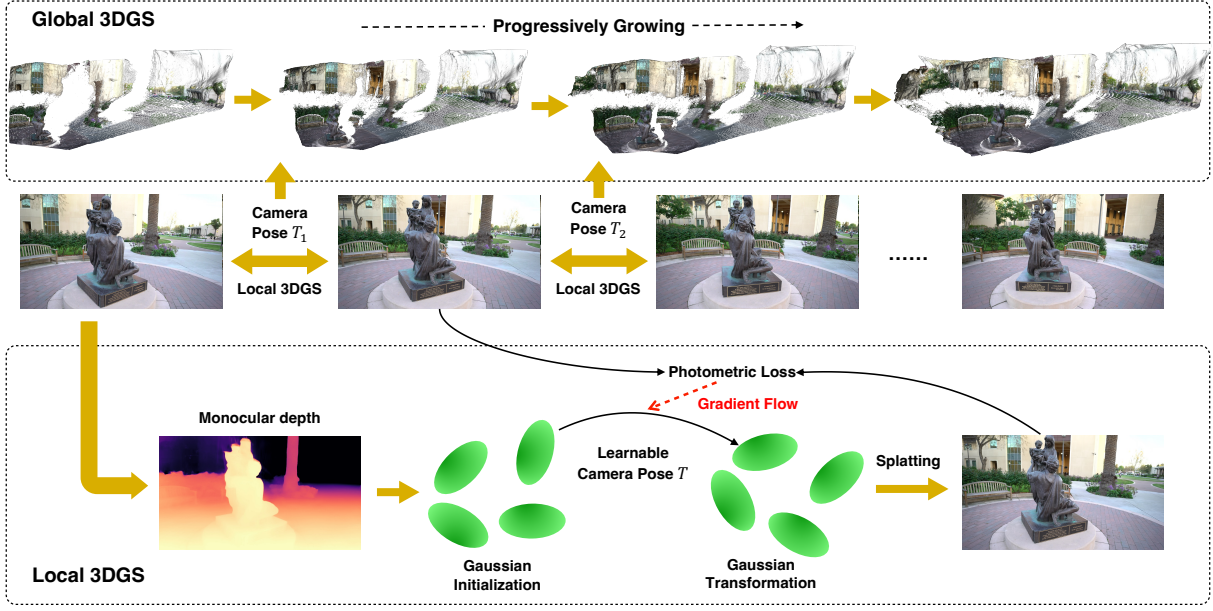


Figure 1.2. Overview of proposed CF-3DGS. Our method takes a sequence of images as input to learn a set of 3D Gaussian that presents the input scene and jointly estimates the camera poses of the frames. We first introduce a local 3DGS to estimate the relative pose of two nearby frames by approximating the Gaussian transformation. Then, a global 3DGS is utilized to model the scene by progressively growing the set of 3D Gaussian as the camera moves.

with the camera pose W yields:

$$\mu_{2D} = K(W\mu)/(W\mu)_z \quad (1.5)$$

where K is the intrinsic projection matrix. Alternatively, the 2D projection μ_{2D} can be obtained from the orthogonal direction $\mathbb{I}$ of a set of rigidly transformed points, i.e., $\mu' = W\mu$, which yields $\mu_{2D} := K(\mathbb{I}\mu')/(\mathbb{I}\mu')_z$. As such, estimating the camera poses W is equivalent to estimating the transformation of a set of 3D Gaussian points. Based on this finding, we designed the following algorithm to estimate the relative camera pose.

Initialization from a single view. As demonstrated in Fig. 1.2 (bottom part), given a frame I_t at timestep t , we first utilize an off-the-shelf monocular depth network, *i.e.*, DPT [120], to generate the monocular depth, denoted as D_t . Given that monocular depth D_t offers strong geometric cues without needing camera parameters, we initialize 3DGS with points lifted from

monocular depth, leveraging camera intrinsic and identity camera pose, instead of the original SfM points. After initialization, we learn a set of 3D Gaussian G_t with all attributes to minimize the photometric loss between the rendered image and the current frame I_t ,

$$G_t^* = \arg \min_{c_t, r_t, s_t, \alpha_t} \mathcal{L}_{rgb}(\mathcal{R}(G_t), I_t), \quad (1.6)$$

where $\mathcal{R}$ is the 3DGS rendering process. The photometric loss $\mathcal{L}_{rgb}$ is $\mathcal{L}_1$ combined with a D-SSIM:

$$\mathcal{L}_{rgb} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{D-SSIM} \quad (1.7)$$

We use $\lambda = 0.2$ for all experiments. This step is quite lightweight to run and only takes around 5s to fit the input frame I_t .

Pose Estimation by 3D Gaussian Transformation. To estimate the relative camera pose, we transform the pre-trained 3D Gaussian G_t^* by a learnable SE-3 affine transformation T_t into frame $t + 1$, denoted as $G_{t+1} = T_t \odot G_t$. The transformation T_t is optimized by minimizing the photometric loss between the rendered image and the next frame I_{t+1}

$$T_t^* = \arg \min_{T_t} \mathcal{L}_{rgb}(\mathcal{R}(T_t \odot G_t), I_{t+1}), \quad (1.8)$$

Note that during the aforementioned optimization process, we freeze all attributes of the pre-trained 3D Gaussian G_t^* to separate the camera movement from the deformation, densification, pruning, and self-rotation of the 3D Gaussian points. The transformation T is represented in form of quaternion rotation $\mathbf{q} \in \mathfrak{so}(3)$ and translation vector $\mathbf{t} \in \mathbb{R}^3$. As two adjunct frames are close, the transformation is relatively small and easier to optimize. Similar to the initialization phase, the pose optimization step is also quite efficient, typically requiring only 5-10 seconds.

1.3.2 Global 3DGS with Progressively Growing

By employing the local 3DGS on every pair of images, we can infer the relative pose between the first frame and any frame at timestep t . However, these relative poses could be noisy resulting in a dramatic impact on optimizing a 3DGS for the whole scene. To tackle this issue, we propose to learn a global 3DGS progressively in a sequential manner.

As described in the top part of Fig. 1.2, starting from the t th frame I_t , we first initialize a set of 3D Gaussian points with the camera pose set as orthogonal, as aforementioned. Then, utilizing the local 3DGS, we estimate the relative camera pose between frames I_t and I_{t+1} . Following this, the global 3DGS updates the set of 3D Gaussian points, along with all attributes, over N iterations, using the estimated relative pose and the two observed frames as inputs. As the next frame I_{t+2} becomes available, this process is repeated: we estimate the relative pose between I_{t+1} and I_{t+2} , and subsequently infer the relative pose between I_t and I_{t+2} .

To update the global 3DGS to cover the new view, we densify the Gaussians that are "under-reconstruction" as new frames arrive. As suggested in [60], we determine the candidates for densification by the average magnitude of view-space position gradients. Intuitively, the unobserved frames always contain regions that are not yet well reconstructed, and the optimization tries to move the Gaussians to correct with a large gradient step. Therefore, to make the densification concentrate on the unobserved content/regions, we densify the global 3DGS every N steps that aligns with the pace of adding new frames. In addition, instead of stopping the densification in the middle of the training stage, we keep growing the 3D Gaussian points until the end of the input sequence. By iteratively applying both local and global 3DGS, the global 3DGS will grow progressively from the initial partial point cloud to the completed point cloud that covers the whole scene throughout the entire sequence, and simultaneously accomplish photo-realistic reconstruction and accurate camera pose estimation.

1.4 Experiments

1.4.1 Experimental Setup

Datasets. We conduct extensive experiments on different real-world datasets, including Tanks and Temples [68] and CO3D-V2 [123]. **Tanks and Temples:** Similar to [9], we evaluate novel view synthesis quality and pose estimation accuracy on 8 scenes covering both indoor and outdoor scenes. For each scene, we sample 7 images from every 8-frame clip as training samples and test the novel view synthesis quality on the remaining 1/8 images. The camera poses are estimated and evaluated on all training samples after Umeyama alignment [151]. **CO3D-V2:** It consists of thousands of object-centric videos where the whole object is kept in view while moving a full circle around it. Compared with Tanks and Temples, recovering camera poses from CO3D videos is much harder as it involves large and complicated camera motions. We randomly select 5 scenes of different categories objects and follow the same protocol to split the train/test set.

Metrics. We evaluate the tasks of novel view synthesis and camera pose estimation. For novel view synthesis, we For camera pose estimation, we report the camera rotation and translation error, including the Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) as in [80, 9]. For novel view synthesis, we report the standard evaluation metrics including PSNR, SSIM [156], and LPIPS [188].

Implementation Details. Our implementation is primarily based on the PyTorch [108] framework and we follow the optimization parameters by the configuration outlined in the 3DGS [60] unless otherwise specified. Notably, we set the number of steps of adding new frames equal to the intervals of point densification, in order to achieve progressive growth of the whole scene. Further, we keep resetting the opacity until the end of the training process, which enables us to integrate the new frames into the Gaussian model established from observed frames. Moreover, the camera poses are optimized in the representation of quaternion rotation $\mathbf{q} \in \mathfrak{so}(3)$ and translation vector $\mathbf{t} \in \mathbb{R}^3$. The initial learning rate is set to 10^{-5} and gradually decay to 10^{-6}

Table 1.1. Novel view synthesis results on Tanks and Temples. Each baseline method is trained with its public code under the original settings and evaluated with the same evaluation protocol. The best results are highlighted in bold.

scenes	Ours			Nope-NeRF			BARF			NeRFmm			SC-NeRF		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Church	30.23	0.93	0.11	25.17	0.73	0.39	23.17	0.62	0.52	21.64	0.58	0.54	21.96	0.60	0.53
Barn	31.23	0.90	0.10	26.35	0.69	0.44	25.28	0.64	0.48	23.21	0.61	0.53	23.26	0.62	0.51
Museum	29.91	0.91	0.11	26.77	0.76	0.35	23.58	0.61	0.55	22.37	0.61	0.53	24.94	0.69	0.45
Family	31.27	0.94	0.07	26.01	0.74	0.41	23.04	0.61	0.56	23.04	0.58	0.56	22.60	0.63	0.51
Horse	33.94	0.96	0.05	27.64	0.84	0.26	24.09	0.72	0.41	23.12	0.70	0.43	25.23	0.76	0.37
Ballroom	32.47	0.96	0.07	25.33	0.72	0.38	20.66	0.50	0.60	20.03	0.48	0.57	22.64	0.61	0.48
Francis	32.72	0.91	0.14	29.48	0.80	0.38	25.85	0.69	0.57	25.40	0.69	0.52	26.46	0.73	0.49
Ignatius	28.43	0.90	0.09	23.96	0.61	0.47	21.78	0.47	0.60	21.16	0.45	0.60	23.00	0.55	0.53
mean	31.28	0.93	0.09	26.34	0.74	0.39	23.42	0.61	0.54	22.50	0.59	0.54	23.76	0.65	0.48

until convergence. All experiments are conducted on a single RTX 3090 GPU.

1.4.2 Comparing with Pose-Free Methods

In this subsection, we compare our method with several baselines including, Nope-NeRF [9], BARF [80], NeRFmm [157] and SC-NeRF [59] on both novel view synthesis and camera pose estimation.

Novel View Synthesis. Different from the standard setting where the camera poses of testing views are given, we need to first obtain the camera poses of test views for rendering. Inspired by NeRFmm [157], we freeze the pre-trained 3DGS model that trained on the training views, while optimizing testing views’ camera poses via minimizing the photometric error between the synthesised images and the test views. To speed up convergence, the camera pose of each test view is initialised by the closest camera position from the learnt camera poses of all training frames, which are then fine-tuned with the photometric error to obtain the testing camera pose. The same procedure is performed on all baselines for a fair comparison.

We report the comparison results on Tanks and Temples in Table 1.1. Our method consistently outperforms the others across all metrics. Notably, compared to Nope-NeRF [9] which takes a significantly longer training time, our approach achieves superior results within a shorter training duration (*e.g.*, 25 hrs vs. 1.5 hrs). We also show the qualitative results in Fig. 1.3. As illustrated in Fig. 1.3, The images synthesized through our approach are significantly sharper and clearer than those produced by the other methods, as evidenced by the notably higher

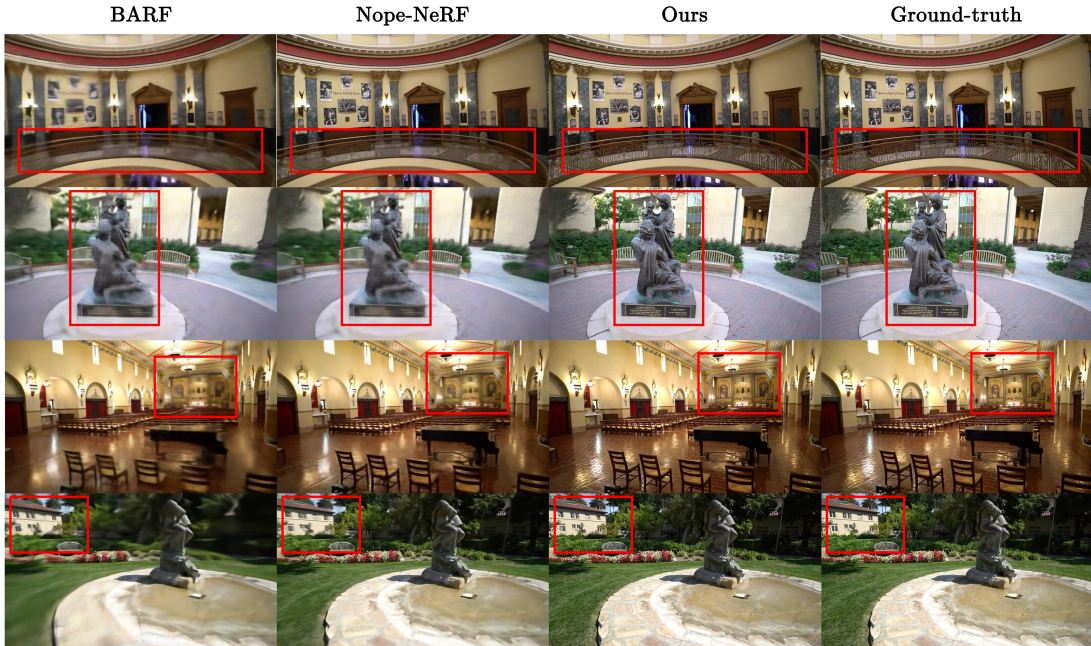


Figure 1.3. Qualitative comparison for novel view synthesis on Tanks and Temples. Our approach produces more realistic rendering results than other baselines. **Better viewed when zoomed in.**

scores in terms of SSIM and LPIPS, as shown in Table 1.1.

Camera Pose Estimation. The learnt camera poses are post-processed by Procrustes analysis as in [80, 9] and compared with the ground-truth poses of training views. The quantitative results of camera pose estimation are summarized in Table 1.2. Our approach achieves comparable performance with the current state-of-the-art results. We hypothesize that the relatively poorer performance in terms of RPE_r and RPE_t may be attributed to relying solely on photometric loss for relative pose estimation in a local region. In contrast, Nope-NeRF incorporates additional constraints on relative poses beyond photometric loss, including the chamfer distance between two point clouds. As indicated in [9], omitting the point cloud loss leads to a significant decrease in pose accuracy.

Table 1.2. Pose accuracy on Tanks and Temples. Note that we use COLMAP poses in Tanks and Temples as the “ground truth”. The unit of RPE_r is in degrees, ATE is in the ground truth scale and RPE_t is scaled by 100. The best results are highlighted in bold.

scenes	Ours			Nope-NeRF			BARF			NeRFmm			SC-NeRF		
	$RPE_t \downarrow$	$RPE_r \downarrow$	ATE $\downarrow$	RPE_t	RPE_r	ATE	RPE_t	RPE_r	ATE	RPE_t	RPE_r	ATE	RPE_t	RPE_r	ATE
Church	0.008	0.018	0.002	0.034	0.008	0.008	0.114	0.038	0.052	0.626	0.127	0.065	0.836	0.187	0.108
Barn	0.034	0.034	0.003	0.046	0.032	0.004	0.314	0.265	0.050	1.629	0.494	0.159	1.317	0.429	0.157
Museum	0.052	0.215	0.005	0.207	0.202	0.020	3.442	1.128	0.263	4.134	1.051	0.346	8.339	1.491	0.316
Family	0.022	0.024	0.002	0.047	0.015	0.001	1.371	0.591	0.115	2.743	0.537	0.120	1.171	0.499	0.142
Horse	0.112	0.057	0.003	0.179	0.017	0.003	1.333	0.394	0.014	1.349	0.434	0.018	1.366	0.438	0.019
Ballroom	0.037	0.024	0.003	0.041	0.018	0.002	0.531	0.228	0.018	0.449	0.177	0.031	0.328	0.146	0.012
Francis	0.029	0.154	0.006	0.057	0.009	0.005	1.321	0.558	0.082	1.647	0.618	0.207	1.233	0.483	0.192
Ignatius	0.033	0.032	0.005	0.026	0.005	0.002	0.736	0.324	0.029	1.302	0.379	0.041	0.533	0.240	0.085
mean	0.041	0.069	0.004	0.080	0.038	0.006	1.046	0.441	0.078	1.735	0.477	0.123	1.890	0.489	0.129

Table 1.3. Novel view synthesis results on CO3D V2. Each baseline method is trained with its public code under the original settings and evaluated with the same evaluation protocol. The best results are highlighted in bold.

Method	Times $\downarrow$	110_13051_23361			415_57112_110099			106_12648_23157			245_26182_52130			34_1403_4393		
		PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Nope-NeRF [9]	~30 h	26.86	0.73	0.47	24.78	0.64	0.55	20.41	0.46	0.58	25.05	0.80	0.49	28.62	0.80	0.35
Ours	~2 h	29.69	0.89	0.29	26.21	0.73	0.32	22.14	0.64	0.34	27.24	0.85	0.30	27.75	0.86	0.20

1.4.3 Results on Scenes with Large Camera Motions

Given that the camera motion in scenes from the Tanks and Temples dataset is relatively small, we further demonstrate the robustness of our approach by validating it on the CO3D videos, which present more complex and challenging camera movements. We first evaluate the quality of synthesised images following the same evaluation procedure used in Tanks and Temples data. As demonstrated in Table 1.3, for novel view synthesis, our approach also significantly outperforms Nope-NeRF which corroborates the conclusions drawn from experiments conducted on the Tanks and Temples dataset. More qualitative results are shown in Fig. 1.4.

In addition, we evaluated camera pose estimation on the CO3D V2 dataset with the provided ground-truth poses for reference. As detailed in Table 1.4, different from the on-par results on Tanks and Temples, our approach consistently surpasses Nope-NeRF across all metrics by a large margin when testing on CO3D V2. This enhanced performance demonstrates the robustness and accuracy of our proposed method in estimating camera poses, especially in scenarios with complex camera movements.

Table 1.4. Pose accuracy on CO3D V2. Note that the camera poses provided by CO3D as the “ground truth”. The unit of RPE_r is in degrees, ATE is in the ground truth scale and RPE_t is scaled by 100. The best results are highlighted in bold.

Method	Times ↓	110_13051_23361			415_57112_110099			106_12648_23157			245_26182_52130			34_1403_4393		
		RPE_t ↓	RPE_r ↓	ATE ↓	RPE_t	RPE_r	ATE	RPE_t	RPE_r	ATE	RPE_t	RPE_r	ATE	RPE_t	RPE_r	ATE
Nope-NeRF [9]	~30 h	0.400	1.966	0.046	0.326	1.919	0.054	0.387	1.312	0.049	0.587	1.867	0.038	0.591	1.313	0.053
Ours	~2 h	0.140	0.401	0.021	0.110	0.424	0.014	0.094	0.360	0.008	0.239	0.472	0.017	0.505	0.211	0.009

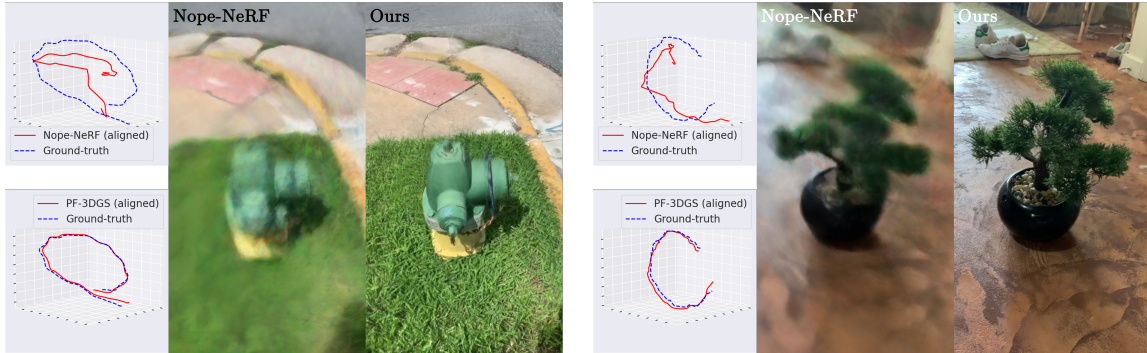


Figure 1.4. Qualitative comparison for novel view synthesis and camera pose estimation on CO3D V2. Our approach estimates camera pose much more robust than Nope-NeRF, and thus generates higher quality rendering images. **Better viewed when zoomed in.**

1.4.4 Ablation Study

In this section, we analyse the effectiveness of different pipeline designs and components that have been added to our approach.

Effectiveness of Progressively Growing. We first validate the effectiveness of progressively growing by removing it from the optimization of the global 3DGS. In other words, we change the current one-stage pipeline to a two-stage process, where the camera pose estimation and 3D Gaussian splatting are learnt in two separate steps. We report the performance on novel view synthesis and camera poses estimation w./w.o progressively growing in Table 1.5. We observe that the progressively growing is essential for enhancing both novel view synthesis and pose estimation. Without progressive growth, 3DGS is unable to utilize the continuity present in videos, which results in the unstable optimization of the global 3DGS model.

RGB Loss vs Depth Loss. Depth-related losses play a crucial role in some advanced pose-unknown approaches, such as Nope-NeRF [9]. To evaluate the significance of depth-related

Table 1.5. Ablation for Progressively Growing on Tanks and Temples. Performance on both novel view synthesis and camera pose estimation. The best results are highlighted in bold.

scenes	w.o. growing				Ours			
	PSNR	SSIM	RPE _t	RPE _r	PSNR	SSIM	RPE _t	RPE _r
Church	22.01	0.72	0.044	0.122	30.23	0.93	0.008	0.018
Barn	25.20	0.85	0.152	0.232	31.23	0.90	0.034	0.034
Museum	20.95	0.70	0.079	0.212	29.91	0.91	0.052	0.215
Family	22.30	0.77	0.065	0.028	31.27	0.94	0.022	0.024
Horse	23.47	0.81	0.147	0.066	33.94	0.96	0.112	0.057
Ballroom	23.36	0.79	0.056	0.073	32.47	0.96	0.037	0.024
Francis	22.20	0.69	0.147	0.161	32.72	0.91	0.029	0.154
Ignatius	21.05	0.67	0.24	0.058	28.43	0.90	0.033	0.032
mean	22.57	0.75	0.116	0.119	31.28	0.93	0.041	0.069

Table 1.6. Ablation study of depth loss on Tanks and Temples. We report the performance on both novel view synthesis and camera pose estimation. The best results are highlighted in bold.

scenes	w. depth				Ours			
	PSNR	SSIM	RPE _t	RPE _r	PSNR	SSIM	RPE _t	RPE _r
Church	28.93	0.91	0.008	0.017	30.23	0.93	0.008	0.018
Barn	28.70	0.87	0.029	0.033	31.23	0.90	0.034	0.034
Museum	26.92	0.83	0.049	0.216	29.91	0.91	0.052	0.215
Family	29.05	0.94	0.021	0.024	31.27	0.94	0.022	0.024
Horse	30.86	0.94	0.108	0.054	33.94	0.96	0.112	0.057
Ballroom	30.38	0.94	0.038	0.018	32.47	0.96	0.037	0.024
Francis	29.97	0.88	0.029	0.154	32.72	0.91	0.029	0.154
Ignatius	26.69	0.87	0.032	0.033	28.43	0.90	0.033	0.032
mean	28.94	0.90	0.039	0.069	31.28	0.93	0.041	0.069

loss, we employed both RGB and depth loss as the objective function during the optimization. As listed in Table 1.6, we observe that the depth loss is not as effective as used in NeRFs. The performance on novel view synthesis is even better when merely using the photometric loss.

Comparison with 3DGS with COLMAP poses. We also compare the novel view synthesis quality of our proposed framework against the original 3DGS [60], which was trained using COLMAP-derived poses on the Tanks and Temples dataset. As indicated in Table 1.7, our joint optimization framework achieves performance comparable to the 3DGS model trained with COLMAP-assisted poses.

Table 1.7. Comparison to 3DGS trained with COLMAP poses. We report the performance of novel view synthesis using ours and vanilla 3DGS. The best results are highlighted in bold

scenes	Ours			COLMAP + 3DGS		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Church	30.23	0.93	0.11	29.93	0.93	0.09
Barn	31.23	0.90	0.10	31.08	0.95	0.07
Museum	29.91	0.91	0.11	34.47	0.96	0.05
Family	31.27	0.94	0.07	27.93	0.92	0.11
Horse	33.94	0.96	0.05	20.91	0.77	0.23
Ballroom	32.47	0.96	0.07	34.48	0.96	0.04
Francis	32.72	0.91	0.14	32.64	0.92	0.15
Ignatius	28.43	0.90	0.09	30.20	0.93	0.08
mean	31.28	0.93	0.09	30.20	0.92	0.10

1.5 Conclusion

In this work, we present CF-3DGS, an end-to-end framework for joint camera pose estimation and novel view synthesis from a sequence of images. We demonstrate that previous works either have difficulty handling large camera motions or require extremely long training durations. Diverging from the implicit representation of NeRFs, our approach utilizes explicit point clouds to represent scenes. Leveraging the capabilities of 3DGS and the continuity inherent in video streams, our method sequentially processes input frames, progressively expanding the 3D Gaussians to reconstruct the entire scene. We show the effectiveness and robustness of our approach on challenging scenes like 360° videos. Thanks to the advantages of Gaussian splatting, our approach achieves rapid training and inference speeds.

Limitations. Our proposed method optimizes camera pose and 3DGS jointly in a sequential manner, thereby restricting its application primarily to video streams or ordered image collections. Exploring extensions of our work to accommodate unordered image collections represents an intriguing direction for future research.

Acknowledgments

Chapter 1, in full, is based on the paper from Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros and Xiaolong Wang, “COLMAP-Free 3D Gaussian Splatting” published

in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. The dissertation author is the primary investigator and author of this paper.

Chapter 2

Scalable 3D Data Curation from Generative Models

2.1 Introduction

Understanding how human hands interact with objects has been a long-standing problem in computer vision. Recently, researchers have tried to scale up such understandings by collecting videos on hand-object interactions [36, 27]. Models trained with these datasets focus on performing hand-object relational reasoning in 2D space. To enable broader applications in robotics and VR/AR, more efforts have been spent on collecting hand-object interaction data with 3D annotations via multiple cameras [13] and new labeling approaches with prepared object CAD models [38]. However, such a data collection process is not scalable and most datasets only contain dozens of objects.

Given the recent advancement of generative modeling with diffusion process [48, 137], can we leverage them to generate realistic 3D hand-object interaction data? While state-of-the-art diffusion models such as Dall-E2 [119] and Stable Diffusion [124] can generate realistic images given text instructions, they still fail quite often when it comes to capturing the details of how fingers are placed around the object. As shown on the left side of Figure 2.1, Stable Diffusion might not be able to output physically or geometrically plausible interactions and sometimes there are more than five fingers in a hand. Moreover, it is still unclear how to configure image outputs beyond text instructions and make the output correspond to 3D shapes and poses.

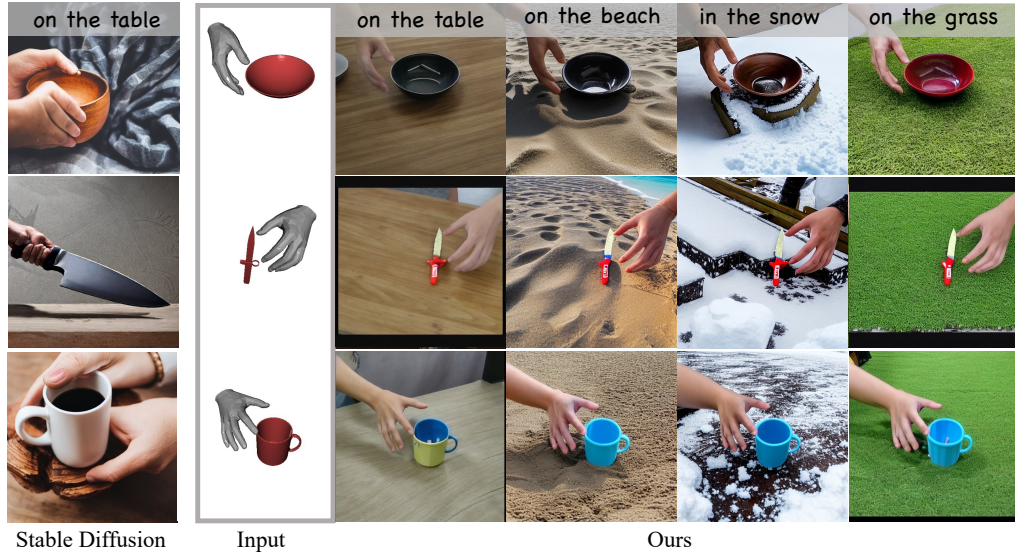


Figure 2.1. **Left:** Hand-object synthesis with Stable Diffusion model; **(ii) Right:** HOIDiffusion generates high-quality hand-object interaction images conditioned on physical structures and detailed text description. The model disentangles the geometry from appearance, exhibiting high generation diversity. Each row: We can fix the structure and control the style based on text inputs; Each column: We can fix the style and control the structure based on 3D structural inputs.

In this paper, we propose to generate 3D hand-object interaction data, i.e., realistic images come with 3D ground-truths at the same time. Beyond realistic generation, we also enable controllable synthesis where the users can specify the geometry configuration and appearance in a disentangled manner. We achieve this by introducing a two-stage framework: We first synthesize the 3D geometric structure (shape and pose) of the hand and the object, and then we train a diffusion model conditioned on both the 3D structure and the text (indicating the style) to synthesize the corresponding RGB image. We visualize some synthesis results on the right side of Figure 2.1. In each row we generate images with the same 3D structural configuration but with different object and background styles; In each column, we fix the styles and synthesize images with different geometric structures.

In the first stage of our framework, we generate a human grasp based on a given 3D object model. We apply the pre-trained GrabNet [142] for this task, which takes the object mesh as inputs for a Variational AutoEncoder and predicts different grasp poses as outputs. In the second

stage, we train a diffusion model conditioning the hand-object geometric configurations. We fine-tune the pre-trained Stable Diffusion model [124] with a few human demonstrations from the DexYCB dataset [13]. We convert the hand-object geometry to estimated surface normals, segmentation, and hand keypoint 2D projection as conditional inputs for the new diffusion model, specifying the structure of the image to generate. The diffusion model will also take text inputs for specifying appearance. During training, we apply a background regularization strategy to reduce the bias brought by DexYCB which comes with the same clean background. The fine-tuned diffusion model leverages both the rich appearance information from the pre-trained model and the geometry information from the new conditional variables.

In our experiments, our method outperforms previous approaches on hand-object image synthesis with more physically plausible interactions. The disentangled design provides flexible control of geometry and appearance. Interestingly, the model shows a strong generalization ability to different text prompts when changing the foreground and background appearance. We use several metrics to evaluate generation fidelity to real datasets and visual alignment with provided prompts. The results show an improvement compared to baselines. Additionally, with a generated dataset with both images and corresponding 3D geometry using our pipeline, we can use it to train an object pose estimator as a downstream application. Our experiments indeed show such realistic synthesized data is very helpful in improving the perception metrics.

2.2 Preliminary: Diffusion Models

Denoising diffusion model [48] is a kind of new generative model with competitive performance and more stable training. It consists of two main processes: diffusion and denoising. In the forward process, randomly sampled noises are added to original images, which can be mathematically simplified as $q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I)$, where α_t is hyperparameters control noise scheduling, and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$. After T steps, the distribution shifted from image space to approximate standard Gaussian distribution. And U-Net model is trained to predict the

added noise. During inference, an image is initialized from the Gaussian distribution and the model removes the noise within T steps, with each step t formulated as

$$\hat{\epsilon}_t = f_\theta(x_t, t) \quad (2.1)$$

where f_θ is the U-Net model. The estimated image at the next time step can be derived and written as

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \alpha_t}} \hat{\epsilon}_t \right) + \sigma_t z \quad (2.2)$$

The simpler version of training loss is: $\|\epsilon - f_\theta(\sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon, t)\|^2$. Therefore diffusion models are able to generate photorealistic images.

2.3 Hand-Object-Interaction Diffusion Model

In order to generate scalable hand-object-interaction images, three expectations need to be satisfied: (i) The model should generate realistic images that are consistent with the geometric description of the specified object. (ii) It should retain the prompt-editing capabilities inherent in the stable diffusion model while incorporating controllable conditions. (iii) The model should have a better generalization ability to synthesize images of unseen instances or categories.

To meet the above requirements, we propose a two-stage approach. For the first stage, our goal is to establish the conditions, primarily the hand-grasping trajectory, for the subsequent stage. Specifically, we utilized a generalizable VAE model trained on large-scale 3D physical data to obtain the ending pose and interpolated the trajectory using spherical linear interpolation. For the second stage, we extracted multiple geometry structures either from the generated grasping trajectory or from images with natural hand-grasping actions through rendering and off-the-shelf estimators to finetune a controllable Stable Diffusion, which enables precise pose control at inference. Furthermore, a background regularization strategy is introduced in our pipeline to mitigate edit ability degradation brought by finetuning. The entire pipeline is shown in Figure

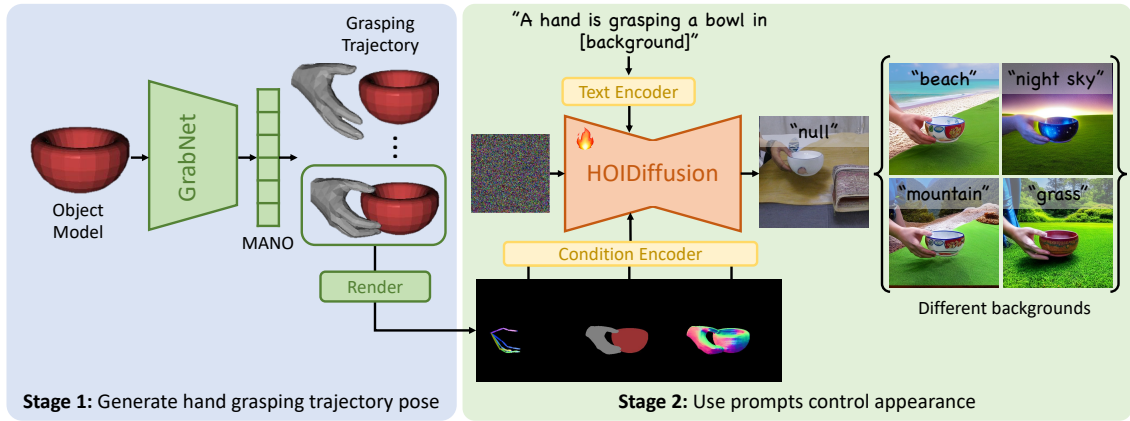


Figure 2.2. HOIDiffusion Pipeline. We propose a two-stage pipeline to synthesize hand-object-interaction data. During the first stage, we utilize a pretrained GrabNet to output 3D hand poses given by a single object model. Then in the second stage, we use those 3D hand poses along with segmentation maps, normal maps and skeletons to conditionally generate high-quality HOI data.

2.2.

2.3.1 Hand Grasping Trajectory Generation

In the first stage of our framework, given a randomly transformed object model, we require a hand-grasping trajectory to effectively reach the object. To this end, we adopted an interpolation method using GrabNet, a model for generating hand grasps conditioned on BPS (Body Part Segmentation) [113] derived from object models. Through extensive training on a large dataset, it has learned to accurately map contact between hands and objects, showcasing strong generalization capabilities for unseen scenarios. Utilizing GrabNet, we can determine the final grasp hand MANO parameters, which include joint pose and hand shape. Given the ending pose and the object’s position, we can approximate an initial starting point—positioned vertically above the contact surface and within a certain distance—using zeroed MANO parameters. Subsequently, we employ spherical linear interpolation between this starting point and the ending pose to create a smooth hand-grasping trajectory. As shown in Figure 2.2 Stage 1, the grasping parameters can be obtained along the trajectory and we could acquire different ground-truth annotations such as segmentation, mask, and depth maps through rendering.

2.3.2 Hand-Object-Interaction image Synthesis

Our Hand-Object-Interaction image synthesis module mainly consists of two components: structure control and appearance regularization, to disentangle the geometry and appearance separately.

Structure Control. Given the hand-grasp trajectory, we extract multiple geometric conditions and leverage the advanced Stable Diffusion models to generate realistic images that are consistent with the given conditions. To precisely control the hand-object image generation, three structure conditions are distilled to control the generation. i) To synthesize realistic hands without missing fingers which is the problem encountered in original Stable Diffusion, hand position information is essentially required. Instead of directly using MANO parameter vectors as guidance, we projected the skeleton information onto the image space as visual control, which can be denoted as s^i . ii) Additionally, to mitigate the inter-disturbance of hand object areas and degradation in performance brought by occlusion, hand-object segmentation (m^i) is used to provide clear boundaries to separate areas, and coarse object shape priors. iii) Finally, we also apply an estimated normal map (n^i) to seize the surface geometry with lighting. The forward process defined in Equation 2.1 can now be defined as $\hat{\epsilon}_t^i = f_\theta(x_t^i, t, [s^i, m^i, n^i])$. With the above controls, the structure information could be disentangled from the appearance, and thus during the inference, we could seamlessly synthesize an accurate image with new poses.

Appearance Regularization. With the above component, we are capable of synthesizing images aligned with diverse condition geometries during inference. However, a notorious drawback of fine-tuning is its tendency to converge or overfit quickly to the training dataset’s style, thereby significantly reducing image diversity. To mitigate this problem and fully harness the capabilities of text-to-image diffusion models for flexible style transformation via prompts, we introduce an appearance regularization method combined with classifier-free guidance [49] as shown in Figure 2.3. Specifically, in addition to using the original Hand-Object Interaction (HOI) training dataset, we synthesize batches of high-quality scenery images with the pre-

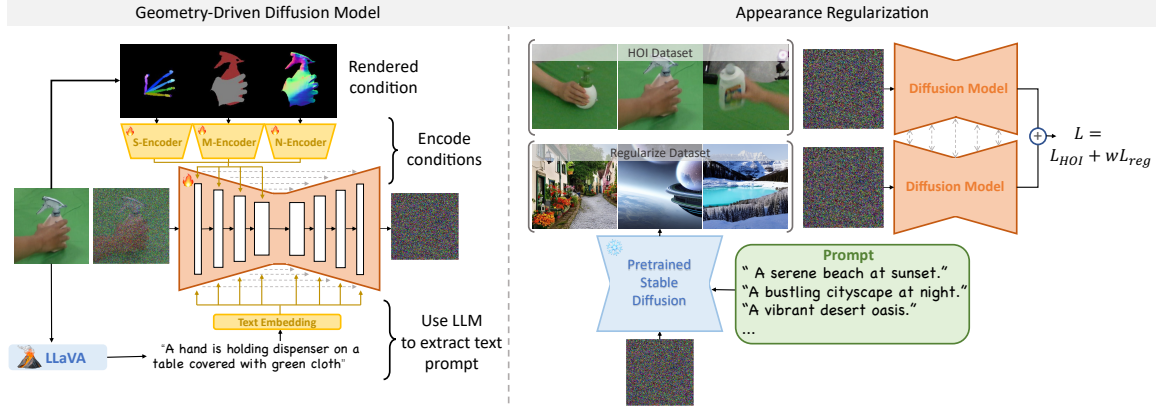


Figure 2.3. HOIDiffusion Architecture. We inject three conditional encoders into the stable diffusion model. We utilize both the HOI datasets and high-quality background images to train HOIDiffusion. The background images are synthesized using the scenery prompts. The texts sent to the model are output by LLaVA for detailed description.

trained text-to-image diffusion model. The prompts for these images are generated by the large language model ChatGPT, forming what we refer to as a “background buffer”. During training, we intermittently utilize these background images for regularization, ensuring it does not detrimentally impact performance. Meanwhile, the paired blank conditions are applied as classifier-free guidance, corresponding to the background region in HOI data. The objective becomes:

$$\begin{aligned} \mathcal{L} = E_{x_0, x_r, \epsilon, \epsilon_r} [& ||\epsilon - f_\theta(\sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon, t)||^2 \\ & + w_r ||\epsilon_r - f_\theta(\sqrt{\alpha_t}x_r + \sqrt{1 - \alpha_t}\epsilon_r, t)||^2] \end{aligned} \quad (2.3)$$

where w_r is the regularization weight, which is set to 1 in our experiments. x_r, ϵ_r are input from the background buffer and corresponding added noise. In addition to the buffer, we also use the large multimodal model LLaVA [85] to caption our training images with detailed descriptions of foreground appearance and background, forcing the model aware of diverse texts with the assistance of the CLIP [117] text encoder.

2.4 Experiment

2.4.1 Implementation Details

To achieve scalable synthesis and improve generalization ability, the training dataset is required to encompass diverse backgrounds and comprehensive annotations. For these purposes, the DexYCB dataset is selected for its high-quality images from varied viewpoints of hand gestures, with human demonstrations and diverse background settings. To prevent overfitting and preserve the text-driven editing ability of diffusion models, we adopt the learning rate of 10^{-5} during training. About 50,000 steps prove sufficient to achieve satisfactory generation quality. When utilizing images from background buffers, we provide vacant skeleton projection and mask conditions, corresponding to the background regions in HOI training data images. This design also can be viewed as a classifier-free guidance that drives the model aware of the condition control effects. The entire training process costs approximately 12 hours on eight A100 GPUs.

2.4.2 Evaluation and Baseline Comparison

Baselines. We compare the results with four baseline models customized for our setting: (1) a fine-tuned LDM [124] without condition modules, working as an unconditional generation task; (2) a modified DreamBooth [127] fine-tuned model, with proposed specific token agnostically applied to all object categories, complemented by regularization data from our trained conditional stable diffusion; (3) Affordance Diffusion [175]; (4) ControlNet with same multiple condition input. It is important to mention that the Affordance Diffusion model is only used for comparison on contact recall evaluation, with quantitative results reported from the original paper.

Image Synthesis Quality. To assess the generation quality, we adopt commonly used metrics FID [46] to evaluate the fidelity of our generated images to real datasets. Additionally, we also present the comparison results with baselines on the Inception Score [128] and sFID [101]. Our reference batch consists of 50k randomly selected images from the training dataset in total.

Table 2.1. Quantitative comparison with previous baseline methods. All models are trained on the DexYCB. We use FID to directly measure the synthesis quality of generated hand-object interaction images. sFID is a recently proposed metric to evaluate image quality using higher-level spatial features. IS is measured for diversity and CLIPScore is to evaluate generated images alignment with provided prompts.

Method	FID↓	sFID↓	IS↑	CLIPScore ↑
LDM [124]	63.71	119.10	6.81	0.68
DreamBooth [127]	134.40	92.82	7.99	0.75
ControlNet [186]	87.99	248.56	6.60	0.77
HOIDiffusion (Ours)	55.22	91.28	7.73	0.78

For the sample batch, we synthesize hand-grasping images corresponding to all randomly rotated object models, some of which are unseen during training. The total sample size is 5k. We refer readers to Table 2.1 for a comprehensive overview of the comparison.

Appearance Alignment. Furthermore, to demonstrate the flexible control over object and background appearance, we use another metrics CLIPScore [45] originally designed for image captioning to automatically evaluate the alignment level between generated images and corresponding prompts. We randomly sample 50 instances from all object models with fixed structure conditions. Multiple appearance descriptions generated by ChatGPT are applied to each instance. The results are shown in Table 2.1. Notably, our method demonstrates superior overall quality of generated images, with higher fidelity to real data and improved alignment with appearance-controlling texts.

Hand Pose Evaluation. For hand-object-interaction images, hand grasping status, and pose precision are of vital importance for real-world applications. A fundamental criterion for synthesized images is the geometric consistency between our generated hands and the provided 2D skeleton projection, disregarding the depth dimension. On top of that, these grasping images sometimes serve as visual demonstrations for various downstream tasks, requiring the exactly accurate contact status between the hand and object in the ending pose image along the grasping trajectory. Consequently, our model is evaluated with the baselines on two key perspectives:

Table 2.2. Evaluation metrics from hand perspective. Hand Contact Recall is to evaluate whether the hand-ending pose is in close contact with the object. PCK is used to measure the accuracy of generated images’ keypoints.

Method	Hand Contact Recall %					PCK↑
	Mug	Bowl	Bottle	Can	Mean↑	
LDM [124]	79.12	78.89	73.00	60.64	72.91	0.15
DreamBooth [127]	79.12	62.22	78.00	75.53	73.72	0.10
ControlNet [186]	86.00	91.40	89.00	93.00	89.85	0.67
Affordance Diffusion [175]	73.00	90.00	90.00	-	84.33	-
HOIDiffusion (Ours)	92.31	97.78	94.00	97.87	95.49	0.85



Figure 2.4. Qualitative results on different structures. Generated images with the same background description but different physical conditions (object shape, poses, and hand skeletons). With plain prompts, HOIDiffusion could generate more realistic images similar to the style in training datasets.

Hand contact recall and hand re-inference accuracy. Specifically, we adopt a contact evaluation setup utilized in Affordance Diffusion [175]. An off-the-shelf hand-object detector [130] is used to classify the image’s in-contact status. Furthermore, to evaluate the re-inference accuracy, we estimate the MANO parameters of hands in images through a widely used single-view hand pose estimator [125], from which we derive the predicted hand joint positions. The percentage of correct keypoints (PCK) is used to measure the accuracy of predicted keypoints representing the hand poses in our data. We evaluate the pose precision on a subset of daily objects. As delineated in Table 2.2, benefited from the geometry guidance, our method manifests the capability to synthesize images depicting accurate grasps and firm contacts, outperforming previous methods in achieving higher mean contact recall and PCK.

2.4.3 Geometry and Appearance Disentangle

The most important design in our model is to disentangle the physical geometries from textures. Through this design, we have observed the remarkable ability of our model to control the generation process with novel object shapes and previously unseen text descriptions. During training, our model learns extensive shape priors from masks and normal map conditions, hence acquiring the capability to transfer to unseen object instances seamlessly. Furthermore, our model preserves the robust text editing ability, enabling flexible style transformation over both background and object appearance. In this subsection, we primarily focus on and showcase the qualitative results of structure and style manipulation.

Geometry Manipulation. In Figure 2.4, we present the generated paired images of four daily seen objects: mug, can, bottle, and bowl, given varying instance shapes, poses, and hand skeletons. The left column of each pair is rendered image in 3D space, from which physical conditions are extracted. Normal maps are obtained from an estimated depth provided by the depth estimator MiDaS [121]. The skeleton and segmentation are also concurrently obtained during rendering. Corresponding generated images are displayed in the right column. All provided prompts follow the format: "A hand is grasping a [object]". The results exhibit an overall generation style in a laboratory or stereo environment, consistent with the realistic appearance style in the training dataset. From Figure 2.1, it is also evident that HOI images generated from our model are more closely aligned with the required geometry than baseline methods.

Background and object appearance control. In this section, we explore the text editing ability with fixed geometry. Through background regularization and classifier-free guidance, our model exhibits the ability to depict diverse background contents, retaining control over appearance using text prompts. We investigate the qualitative performance of various text controls under identical physical conditions, shown in Figure 2.5. Each column represents one distinct background description. Notably, the generated images exhibit high fidelity to



Figure 2.5. Synthesized images with diverse background descriptions. In addition to real-style synthesis, our model also allows users to generate according to their preferences such as science fiction or general landscapes.

the provided prompts, maintaining the layout and structure unchanged. This demonstrates the ability of HOIDiffusion to effectively disentangle the appearance from geometry structures, thus enabling flexible style transformation without geometry distortion. This is essential in data construction, ensuring the precise alignment with input geometry and diverse range of visual appearances.

2.4.4 Ablation Study

Two indispensable components of our design are precise structural control and appearance regularization, effectively improving model performance on geometry consistency and diversity.

Structural Control. In our approach, there are three crucial structure conditions provided to the model and we investigate the importance brought separately by these three modules. Results are presented in Table 2.3. Essentially, each condition serves a distinct purpose: the normal map guides the model in perceiving surface textures with lighting, which is essential for maintaining geometry consistency; hand keypoint projection precisely depicts the pose of hand

Table 2.3. Ablation study on structural control. FID evaluation on 1,000 images of different types of missing modules to demonstrate the necessity of all physical conditions. Our method outperforms all others.

Method	FID (1k) ↓
LDM (finetuned)	86.12
w/o estimated normal maps	82.40
w/o hand keypoint projection	81.93
w/o hand-object segmentation	78.57
HOIDiffusion(Ours)	77.64

Table 2.4. CLIPScore evaluation. Consistency is evaluated between provided prompts and generated images for different backgrounds and instances.

Method	CLIPScore ↑
w/o regularization	0.66
HOIDiffusion(Ours)	0.79

joints, preventing the model from synthesizing multiple fingers or distorted hands; hand-object segmentation provides a clear boundary between different regions, avoiding interference between hand and object areas. As presented in the results, our full model outperforms other incomplete versions in quantitative evaluation.

Appearance Regularization. Table 2.4 and Figure 2.6 demonstrate the significance of appearance regularization in our module. Without regularization, the finetuned model quickly converges to the style in training datasets, mostly in a laboratory/studio environment, as depicted in the first line of Figure 2.6, This convergence impairs the model’s ability to generate diverse images, which is essential for data generation. By incorporating appearance regularization using data generated from the pretrained model, HOIDiffusion mitigates the drift to a fixed style and improves the overall text-editing ability.

2.4.5 Applications

Video Generation. The real hand-object interaction datasets often exhibit data in video format, a complete fetching process to the object. Collecting these video clips is a considerable



Figure 2.6. Ablation study on appearance regularization. Different backgrounds are used as prompts with the same geometry conditions to compare the text editing flexibility brought by the regularization module.

challenge. Some datasets [13, 170] comprising almost millions of images only contain no more than 10k videos, thus video data is much more valuable. In experiments, we observe the significant divergence in generated images between adjacent frames despite the similar provided conditions and texts. This divergence results in dramatic flickering in directly concatenated videos. To this end, we leverage zero-shot video generation techniques in the diffusion model to establish inter-consistency among frames. To be more specific, the original self-attention layers in the U-Net are refactored to cross-attention modules between an anchor frame and current frames. This adjustment establishes the awareness of the previous appearance style, ensuring video consistency. In our experiments, we set the middle frame as an anchor, and with all frames attend to both the anchor image and themselves. This approach effectively mitigates the flickering issue, synthesizing relatively smooth hand-grasping trajectories. The video clip samples are shown in Figure 2.7.

Downstream tasks. Another interesting method to evaluate the performance of HOID-iffusion, is to apply it in downstream tasks as a data augmentation method or new data source. In this section, we explore the potential for improving categorical object 6D pose estimation

Table 2.5. Quantitative evaluation on NOCS. We use SPD and DualPoseNet and change the synthesized images in the dataset with our generated images for training. Our performance improve on all metrics with DualPoseNet and all cm metrics with SPD which demonstrates the good quality of our images and can be utilized for downstream tasks.

Method		IoU@25	IoU@50	IoU@75	5°2cm	5°5cm	10°2cm	10°5cm
SPD [100]	Original	82.9	75.3	48.6	17.7	19.9	38.8	48.3
	Ours	82.5	71.1	47.0	21.1	23.2	43.8	54.5
DualPoseNet [82]	Original	84.1	79.7	60.1	28.0	34.3	47.8	64.2
	Ours	90.9	84.1	65.8	29.2	34.4	55.0	66.6



Figure 2.7. Zero-shot video generation of hand grasping trajectory. Images along the same line represent the sequential motion of reaching an object. By leveraging temporal-level cross-attention, the frame flickering problem is mitigated.

tasks. Most models in this task are trained on dataset NOCS [155], consisting of both synthetic and real data with annotations. Some models [149, 16] implicitly predict normalized object coordinate space, and then utilize Umeyama algorithms to parse the transformation matrix. Others [82] explicitly predict the rotation, translation, and scale parameters. Despite differing in module design, all these methods leverage RGB image encoders to obtain the visual features. An interesting observation is that the synthesized images directly rendered from object models appear too artificial, potentially affecting the performance of the RGB encoder. Inspired by this, we substitute the synthesized images with our generated HOI images in the same poses, anticipating the reality brought by our data could help enhance model performance. We exhibit the results in Table 2.5. We choose two representative object pose estimators for evaluation. The “original” model in the table refers to training using an unchanged NOCS dataset, and “our”

model is trained using our mixed data.

2.5 Conclusion

In this work, we propose HOIDiffusion - a novel conditional diffusion framework for generating realistic, diverse 3D hand-object interaction data. By decoupling 3D geometric structures from high-fidelity RGB synthesis, our two-stage pipeline uses physical priors—such as normal maps and hand skeletons—as 3D-aware guidance. This allows for precise, disentangled control over interaction geometry and visual appearance. Experimental results demonstrate that HOIDiffusion significantly outperforms existing baselines in physical plausibility and generalization across diverse prompts. Furthermore, we validated our synthesized data by improving the performance of 6D object pose estimation models in complex scenarios.

Limitations. Despite these advancements, the model does not strictly enforce physical constraints, occasionally resulting in hand-object penetration or "floating" fingers. Additionally, the diversity of generated interactions is limited by the object categories in the training data (*e.g.*, DexYCB).

Acknowledgments

Chapter 2, in full, is based on the paper from Mengqi Zhang, Yang Fu, Zheng Ding, Sifei Liu, Zhuowen Tu, and Xiaolong Wang, "HOIDiffusion: Generating Realistic 3D Hand-Object Interaction Data" published in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024. The dissertation author is the primary investigator and author of this paper.

Chapter 3

Bridging 2D Vision Language Models to the 3D World

3.1 Introduction

Recent advances in Vision-Language Models (VLMs) [81, 86, 145, 2, 6, 23] have significantly improved their capabilities in a wide range of image and video understanding and reasoning tasks. Despite these achievements, current 2D VLMs face substantial limitations in perceiving and reasoning about the 3D world. This constraint stems from their fundamental architecture, which inherently utilizes 2D visual features and relies exclusively on training with images and video data.

Some recent studies attempt to develop the 3D VLMs by integrating 3D encoders to extract features from point clouds as inputs for VLMs. These 3D encoders are typically trained from scratch on 3D point cloud datasets with ground-truth question-answering pairs. However, due to the scarcity of large-scale 3D datasets, the 3D encoders are often less powerful and robust compared to 2D visual encoders, such as CLIP [118], DINOv2 [105], and RADIO [122], which are trained on billions of 2D images. To leverage these powerful 2D foundation models, several approaches [58, 37, 33] utilize 2D segmentation and CLIP features to obtain pixel-aligned semantic features and construct object-level 3D scene graphs for various 3D understanding tasks. Nevertheless, these approaches suffer from computationally intensive per-scene processing and complex implementation pipelines. In contrast, some efforts [201, 192] have focused on

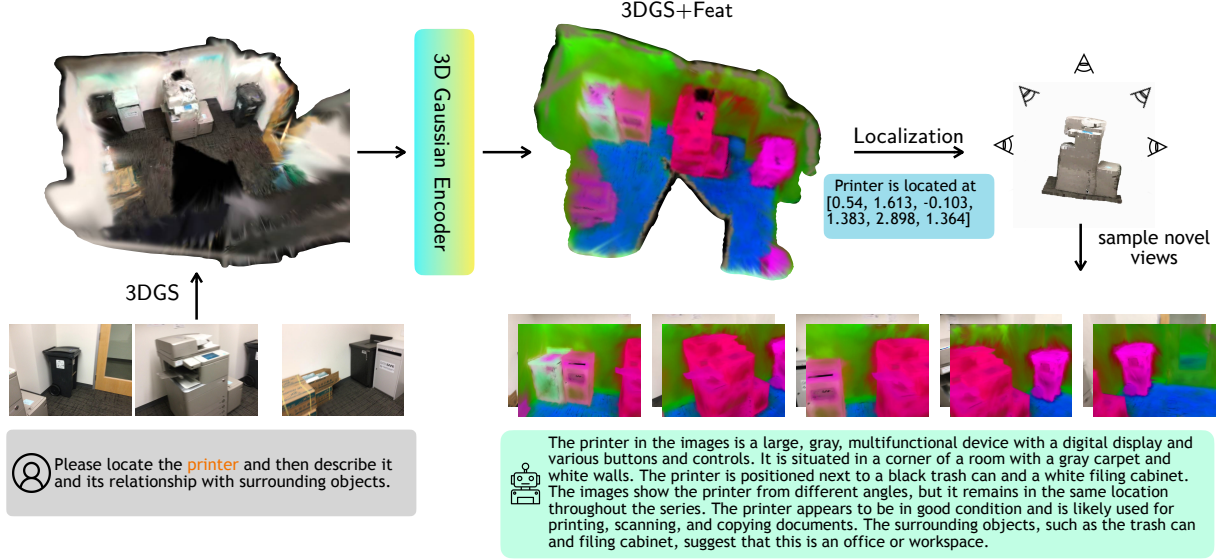


Figure 3.1. SplatDistill Overview. We propose a generalist pipeline leveraging 3D Gaussian Splatting as an intermediate representation to bridge the gap between 2D Vision Language Models (VLMs) and the 3D world for 3D visual understanding. Starting from multi-view images, we first reconstruct the 3D Gaussian Splatting (3DGS) representation (details omitted for simplicity). Our learned 3D Gaussian encoder then generates language-aligned per-Gaussian features from the reconstructed 3D Gaussians. With these feature-enriched 3D Gaussians, regions of interest can be precisely located in 3D space. Furthermore, corresponding feature maps can be efficiently rendered from nearby novel viewpoints and utilized by 2D VLMs, enabling effective 3D scene understanding and reasoning without additional 3D training data.

building 3D VLMs on top of powerful 2D VLMs by incorporating 3D positional information from multi-view RGB-D inputs. While these approaches achieve impressive performance in 3D scene perception and reasoning, they still rely on fine-tuning with multi-view datasets that include ground-truth depth and question-answering pairs. This raises a compelling question: can we enable 2D VLMs to directly perform 3D understanding and reasoning tasks without training on specialized 3D data?

To fully leverage the strengths of 2D VLMs, it is essential to bridge the fundamental gap between the 2D training data and 3D inference environments. In this paper, we propose utilizing 3D Gaussian Splatting (3DGS) [61, 62] as an intermediate representation to connect this gap. 3DGS provides an efficient and interpretable 3D representation with real-time rendering capabilities, making it a strong candidate for seamlessly connecting 2D images to 3D environ-

ments. Building on 3D Gaussian as the scene representation, we aim to map its parameters to a feature space that is well-aligned with several 2D foundation models, enabling compatibility with pretrained VLMs. Recent studies [132, 173, 195, 179] have proposed optimizing an additional attribute as the visual feature alongside 3DGS optimization. However, these per-scene optimization approaches suffer from lengthy processing times and poor generalization to new scenes, limiting their practicality in diverse and complex environments. To address this challenge, we propose a feed-forward framework, *SplatDistill*, to distill the rich language-aligned knowledge from 2D foundation models into a set of 3D Gaussians. Specifically, we introduce a transformer-based network that takes 3D Gaussians, along with their attributes, as inputs to produce high-dimensional features for each Gaussian. The proposed Gaussian transformer is trained using a rendering-and-comparison mechanism: the input Gaussians with extracted features are splatted into 2D space to generate a 2D feature map, which is then compared with the 2D feature map generated by the pretrained 2D foundation model. Therefore, the learned 3D Gaussian features are well-aligned with 2D foundation models and can be directly applied to various downstream tasks, such as 3D segmentation and localization, without requiring task-specific fine-tuning. During inference, these feature-augmented 3D Gaussians are rendered into multiple views, and the resulting feature maps, combined with RGB images, are used as input to pretrained 2D VLMs for 3D visual question answering, as illustrated in Fig. 3.1.

To validate the effectiveness of our approach, we conduct extensive experiments on a wide range of 3D tasks and benchmarks [126, 26, 5, 14, 93] including open-vocabulary scene understanding, 3D visual question answering, dense captioning, and 3D visual grounding. As the Gaussian features are well-aligned with the 2D features through the distillation process, our approach demonstrates improved performance on zero-shot open-vocabulary scene understanding over existing methods that require fine-tuning on each individual scene. Moreover, by connecting 2D VLMs to the 3D Gaussians, our approach largely enhances the 3D spatial understanding and reasoning capabilities of the existing 2D LLMs without fine-tuning on 3D QA ground-truth pairs.

3.2 Method

Given a set of 3DGS reconstructed from a sequence of images along with camera intrinsics and extrinsics, our goal is to learn a distillation network that enables various 3D scene understanding and reasoning tasks. We detail our method in the following sections, starting with a brief review of 3DGS reconstruction in Sec. 3.2.1. Then, we propose to aggregate the pixel features from multiple views onto every Gaussian via multi-view fusion, in Sec. 3.2.2. Next, we introduce a feature distillation framework with a Gaussian encoder using 3D Gaussians as inputs, in Sec. 3.2.3.

3.2.1 3D Gaussian Reconstruction

The first step of our approach is to reconstruct the indoor scenes via 3D Gaussian Splatting [61]. Given a sequence of RGB images with camera intrinsics and extrinsics, we represent a scene with a set of 3D Gaussians as

$$\mathcal{G} = \{(\boldsymbol{\mu}, \text{SH}, \boldsymbol{r}, \boldsymbol{s}, \alpha)_i\}_{i=1:M} \quad (3.1)$$

where (a) $\boldsymbol{\mu} \in \mathbb{R}^3$ is the 3D mean of the Gaussian, (b) $\text{SH} \in \mathbb{R}^{(k+1)^2 \times 3}$ are the spherical harmonics (SH) coefficients that represent the Gaussian color, (c) $\boldsymbol{r} \in \mathbb{R}^4$ is its quaternion rotation factor, (d) $\boldsymbol{s} \in \mathbb{R}^3$ is the Gaussian scale and (e) $\alpha \in \mathbb{R}$ is the Gaussian opacity. Then, the covariance matrix $\boldsymbol{\Sigma}$ describes an ellipsoid configured by a scaling matrix $\boldsymbol{S} = \text{diag}(\boldsymbol{s})$ and rotation matrix $\boldsymbol{R} = \text{q2R}(\boldsymbol{r})$, where $\text{q2R}(\cdot)$ is the formula for constructing a rotation matrix from a quaternion. Then, the covariance matrix can be computed as $\boldsymbol{\Sigma} = \boldsymbol{R}\boldsymbol{S}\boldsymbol{S}^\top\boldsymbol{R}^\top$. The rendered color can be formulated as the alpha-blending of N ordered points that overlap the pixel as follows,

$$\boldsymbol{c}_{\text{pix}} = \sum_i^N \text{SH}_i \alpha_i \prod_j^{i-1} (1 - \alpha_j) \quad (3.2)$$

where α_i represents the density of this Gaussian computed from the per-Gaussian opacity weighted by the Gaussian covariance Σ , which we ignore in Eq. 3.2 for simplicity.

Training Data Setup. To create large amounts of 3DGS reconstruction examples for training the distillation network, we modify the original initialization strategy for better efficiency. In particular, instead of using SfM points, we initialize the Gaussian positions from the ground-truth scene mesh vertices. To reduce the number of initialized Gaussians, we perform the grid subsampling with a voxel size of 2cm on the original mesh. Finally, we remove the Gaussian growing step while only pruning Gaussians during optimization.

3.2.2 Multi-view Feature Fusion

For a given set of 3D Gaussians, our next step is to extract per-Gaussian features. As each scene can contain millions of Gaussians and feature maps from 2D foundation models are typically high-dimensional, directly optimizing 3DGS with high-dimensional per-Gaussian features can be time-consuming and computationally intensive. Therefore, we propose a multi-view feature fusion strategy to lift 2D feature maps into the 3D Gaussian space.

2D Feature Extraction. Previous works [110, 180] typically extract features from task-specific models - for instance, OpenScene utilizes features from a 2D segmentation model specifically for open-vocabulary scene understanding. This approach necessitates multiple model distillations to accomplish different tasks. To address this limitation, we propose to extract backbone features from the RADIO [122] model. Unlike task-specific features, RADIO’s backbone features can be projected into various target embedding spaces (*i.e.*, DINO [105], CLIP [118], SigLIP [183], and SAM [67]) using pretrained lightweight adapters, significantly enhancing our approach’s flexibility.

Multi-view Fusion. For each Gaussian $\mathbf{g}$ from the set of Gaussians $\mathcal{G}$, we first project it onto the image plane using the camera intrinsics K_j and world-to-camera extrinsics E_j of the j th frame. The corresponding pixel coordinate can be calculated as $\mathbf{u} = K_j \cdot E_j \cdot \mathbf{g}(\boldsymbol{\mu})$. (where the homogeneous representations of $\mathbf{u}$ and $\boldsymbol{\mu}$ are omitted for simplicity). We also conduct occlusion

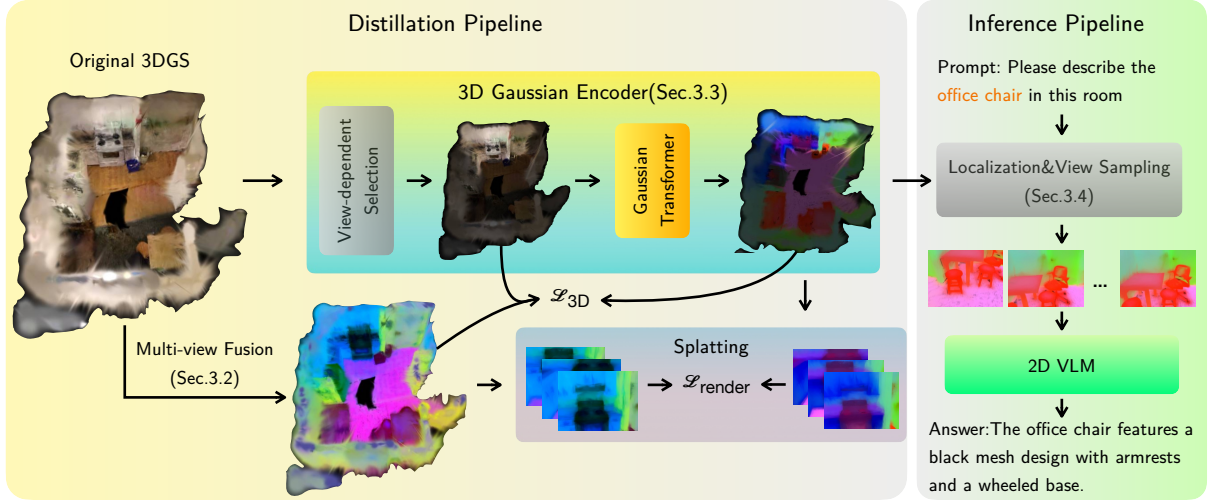


Figure 3.2. Architecture Details of SplatDistill. (Left) The distillation pipeline begins with an original 3DGS reconstructed from multi-view images. Our 3D Gaussian Encoder (Sec.3.2.3) processes these Gaussians through view-dependent selection and a Gaussian Transformer to produce feature-enriched 3D Gaussians. The framework is supervised by two distillation losses: $\mathcal{L}_{3D}$ applied directly to 3D Gaussians via multi-view fusion (Sec. 3.2.2), and $\mathcal{L}_{render}$ applied to 2D rendered feature maps. (Right) The inference pipeline demonstrates how the feature-enriched 3D Gaussians enable object localization and strategic view sampling (Sec.3.2.4) in response to text prompts. The sampled views and their feature maps are fed to a 2D VLM to enable 3D scene understanding without specialized 3D training.

tests to guarantee that only visible Gaussians are considered by comparing them with the rendered depth map. More details about this can be found in the supplementary materials.

Given the projected pixel coordinate $\mathbf{u}$, the corresponding feature can be obtained via $\mathbf{f}_j = \mathbf{F}_j[\mathbf{u}]$, where $\mathbf{F}_j \in \mathbb{R}^{H,W,D}$ is the RADIO backbone feature map and D refers to the feature dimension. Assuming N views are available for fusion, the fused feature vector for Gaussian $\mathbf{g}$ can be computed as the mean of the corresponding features across these N views: $\mathbf{f}_{fused} = \text{mean}(\mathbf{f}_1, \dots, \mathbf{f}_N)$. By repeating this fusion process for each Gaussian, we establish a feature-enriched 3D Gaussian set: $\mathcal{G}_{fused} = \{(\mathbf{f}_{fused}, \boldsymbol{\mu}, \text{SH}, \mathbf{r}, \mathbf{s}, \alpha)_i\}_{i=1:M}$.

3.2.3 3D Feature Distillation

While the per-scene generated 3D feature Gaussians can be directly applied to various downstream tasks, this approach requires repeating the feature extraction and multi-view fusion

procedure for each scene which can take several minutes to process. Moreover, such fused features can be noisy due to potentially inconsistent 2D predictions. Therefore, we propose to distill such knowledge from the 2D foundation model into a 3D network that takes a set of 3D Gaussians as input, as shown in Fig. 3.2.

Specifically, given a set of 3D Gaussians $\mathcal{G}$, we aim to learn a network ϵ_{3D} that outputs per-Gaussian embeddings $\mathbf{F}_{3D} = \epsilon_{3D}(\mathcal{G})$, where $\mathbf{F}_{3D} = \{\mathbf{f}_{3D}^1, \dots, \mathbf{f}_{3D}^M\}$. A strong limitation of learning such a network on 3D Gaussians is the large amounts of Gaussians, *i.e.* millions of Gaussians. Directly feeding these 3D Gaussians into a 3D network like PointTransformer [190] can cause the out-of-memory issue. Motivated by this finding, we explore methods to reduce the number of Gaussians. While a straightforward approach would be to voxelize the Gaussians with larger voxel sizes, this leads to the omission of too many tiny Gaussians, resulting in rendering artifacts. On the other hand, we observe that rendering any specific view typically utilizes only a small portion of the scene’s 3D Gaussians. Based on this observation, we propose a view-dependent selection strategy that further reduces the input Gaussians after conventional voxelization. Specifically, we first sample several target views and construct view frustums from their camera extrinsics. We then filter out Gaussians outside these view frustums, using the remaining Gaussians as input for the 3D network. We utilize PointTransformer V3 [162] as our 3D backbone network ϵ_{3D} and change its output dimension to D .

To ensure consistency between output features and fused features, we employ cosine similarity loss as suggested by [110]. Additionally, to enable pre-trained adapter heads to serve as drop-in replacements for mapping backbone features to different target embedding spaces, we maintain the magnitude of feature vectors using smooth L1 loss [34]. Thus, we employ a combination of cosine similarity and smooth L1 as,

$$\begin{aligned} \mathcal{L}_{\text{match}}(x, y) &= \alpha(1 - \cos(x, y)) + \beta \mathcal{L}_{\text{L1-smooth}}(x, y) \\ \mathcal{L}_{3D} &= \sum_i^M \mathcal{L}_{\text{match}}(\mathbf{f}_{3D}^i, \mathbf{f}_{\text{fused}}^i) \end{aligned} \tag{3.3}$$

where α, β are two balance factors. We empirically set $\alpha = 0.9$ and $\beta = 0.1$. To further ensure that the distilled features remain compatible with pretrained RADIO adapters, we apply the same matching loss between the 2D rendered feature maps and the RADIO 2D features after processing both through pretrained adapter ϕ_h ,

$$\mathcal{L}_{\text{render}} = \sum_h^H \sum_j^N \mathcal{L}_{\text{match}}(\phi_h(\mathcal{R}(\mathcal{G}_{\text{fused}}, K_j, E_j)), \phi_h(\mathbf{F}_j)) \quad (3.4)$$

where $\mathcal{R}$ is the feature rendering process from the given camera parameters and H is the number of adapter heads. Therefore, the final loss function is expressed as,

$$\mathcal{L}_{\text{distill}} = \lambda_{3\text{D}} \mathcal{L}_{3\text{D}} + \lambda_{\text{render}} \mathcal{L}_{\text{render}} \quad (3.5)$$

where $\lambda_{3\text{D}}$ and λ_{render} are set to 1.0, 0.2, respectively.

3.2.4 Inference

Unlike inference with traditional 2D VLMs, which typically requires uniform sampling of multiple frames from a testing sequence, we design a two-stage inference pipeline that leverages the feature-enriched 3DGS to efficiently “focus” on regions of interest. As demonstrated in Fig. 3.2 (right side), given the set of Gaussians with distilled features, we first identify a small subset of Gaussians that covers the content of interest. The distilled features naturally reside in the same language-aligned embedding space as RADIO, enabling us to evaluate semantic similarity between per-Gaussian features and arbitrary text prompts through cosine similarity computation. Subsequently, we strategically sample multiple viewpoints around these identified regions to generate both rendered images and their corresponding feature maps via Gaussian splatting. These visual inputs serve as the foundation for 2D VLM inference, allowing the model to reason effectively about the 3D scene from optimal perspectives.

3.3 Experiments

3.3.1 Experimental Setup

Datasets and Benchmarks. Our training and evaluation experiments are conducted on the ScanNet dataset [26], an extensive indoor 3D scene dataset containing 1,513 multi-modal indoor scenes where 1,201 scenes are utilized for training and 312 scenes are for validation. We conduct our evaluation across several benchmarks adhering to these training/validation splits, including ScanNetV2/ ScanNet200 [26, 126] for 3D scene understanding, ScanQA [5] for 3D QA, Scan2Cap [24] for 3D dense captioning, and ScanRefer [14] for 3D visual grounding.

Implementation Details. To prepare the training data, we first utilize the 3DGS reconstruction pipeline provided in Gsplat studio [174] to generate the 3DGS representation for each scene in ScanNet. To facilitate the optimization process, we instead initialize the Gaussian positions with the subsampled points from the given scene mesh vertex. For the feature distillation, we utilize the RADIOv2.5 ViT-H/16 [122] as the 2D foundation model to extract the 2D feature map from each input RGB frame for its higher resolution output. The Gaussian encoder is trained on 8×A100 80G GPUs for 100 epochs on the training scenes in ScanNetV2. During the inference, without the access of ground-truth meshes, we reconstruct the 3D Gaussians via LoopSplat [202]. To achieve the zero-shot scene segmentation, we directly utilize the distilled Gaussian feature. Meanwhile, for 3D QA tasks, we render these 3D Gaussians along with features to obtain multiple-view feature maps and feed into the pre-trained 2D VLMs. In the following experiments, we utilize pre-trained Llama-3-VILA1.5-8B [81] as the VLM backend by default unless specified.

In our experiments, we evaluate two variants of our approach based on how to obtain 3D feature Gaussians. “Ours (Fused)” refers to features obtained by multi-view fusion (Sec. 3.2.2) taking around 5-10 minutes, and “Ours (Distilled)” denotes features generated in a single forward pass through the 3D Gaussian network (Sec. 3.2.3) within 10 seconds.

Table 3.1. Quantitative comparison with SOTA models for 3D object recognition on ScanNetv2. We report the top-1 classification accuracy.

Method	Avg.	Bed	Cab	Chair	Sofa	Tabl	Door	Wind	Bksf	Pic	Cntr	Desk	Curt	Fridg	Bath	Showr	Toil	Sink
PointCLIP [187]	6.3	0.0	0.0	0.0	0.0	0.7	0.0	0.0	91.8	0.0	0.0	0.0	15.0	0.0	0.0	0.0	0.0	0.0
PointCLIP V2 [204]	11.0	0.0	0.0	23.8	0.0	0.0	0.0	0.0	7.8	0.0	90.7	0.0	0.0	0.0	0.0	64.4	0.0	0.0
CLIP2Point [55]	24.9	20.8	0.0	85.1	43.3	26.5	69.9	0.0	20.9	1.7	31.7	27.0	0.0	1.6	46.5	0.0	22.4	25.6
PointCLIP w/ TP.	26.1	0.0	55.7	72.8	5.0	5.1	1.7	0.0	77.2	0.0	0.0	51.7	0.3	0.0	0.0	40.3	85.3	49.2
CLIP2Point w/ TP.	35.2	11.8	3.0	45.1	27.6	10.5	61.5	2.6	71.9	0.3	33.6	29.9	4.7	11.5	72.2	92.4	86.1	34.0
CLIP ² [182]	38.5	32.6	67.2	69.3	42.3	18.3	19.1	4.0	62.6	1.4	12.7	52.8	40.1	9.1	59.7	41.0	71.0	45.5
Uni3D [194]	45.8	58.5	3.7	78.8	83.7	54.9	31.3	39.4	70.1	35.1	1.9	27.3	94.2	13.8	38.7	10.7	88.1	47.6
OpenIns3D [56]	60.8	85.2	27.4	87.6	77.3	46.9	54.8	64.2	71.4	9.9	80.8	82.7	71.6	61.4	38.7	0.0	87.9	85.7
Ours (Distilled)	70.7	81.2	52.4	83.1	87.5	37.7	84.6	53.9	67.6	9.8	56.0	89.3	55.0	56.5	97.7	95.8	96.4	100.0

3.3.2 Evaluation on Scene Understanding

To evaluate the quality of distilled 3D Gaussian features, we conduct experiments under ScanNet to explore the performance of zero-shot recognition and open-vocabulary instance segmentation.

3D Object Recognition. 3D object recognition aims to recognize the category of the query 3D instance. We compare our approach among all state-of-the-art (SOTA) 3D recognition models: PointCLIP (v1 and v2) [187, 204], CLIP² [182], CLIP2Point [55] and Uni3D [25]. We follow the same setting as CLIP² where the ground truth instance segmentation is provided for all the methods. As these methods do not use any scene from ScanNet during training, for a fair comparison, we also compare our approach with OpenIns3D where a per-scene fusion strategy is utilized to map 2D segmentations into 3D space (note that no annotations are used). All results are summarized in Table 3.1 where our approach produces much more accurate recognition results than existing works.

Open-vocabulary 3D Instance Segmentation. To further show the effectiveness of the distilled Gaussian feature, we adopt the distilled Gaussian feature to open-vocabulary 3D instance segmentation on ScanNet200. We consider three groups of existing SOTA methods: (i) the fully supervised models that utilize the closed-set classification annotations, including ISBNNet [102] and Mask3D [51]. (ii) the per-scene optimization approaches that utilize the class-agnostic 3D mask proposals and lift up the 2D feature/segmentation map from 2D foundation models, *i.e.*, CLIP, into 3D space via some multi-view fusion strategies, *i.e.*, OpenScene [110](using 2D fusion

Table 3.2. Quantitative comparison with SOTA models on ScanNet200 for 3D instance segmentation. Metrics are respectively: AP averaged over an overlapping range, and AP evaluated at 50% and 25% overlaps. We also report AP scores for head, common, and tail subsets of ScanNet200.

Model	AP ₂₅	AP ₅₀	AP	AP _{Head}	AP _{Common}	AP _{Tail}
<i>Closed-vocabulary, fully supervised</i>						
ISBNNet [102]	37.6	32.7	24.5	38.6	20.5	12.5
Mask3D [14]	36.2	41.4	26.9	39.8	21.7	17.9
<i>Open-vocabulary, per-scene</i>						
OpenScene [110](Fused) + Mask3D	17.8	15.2	11.7	13.4	11.6	9.9
OpenMask3D [143]	23.1	19.9	15.4	17.1	14.1	14.9
Open3DIS* [103]	27.3	23.1	18.6	24.7	16.9	13.3
OpenIns3D [56]	14.4	10.3	8.8	16.0	6.5	4.2
Ours (Fused)	28.1	24.3	19.1	22.6	18.6	11.1
<i>Open-vocabulary, feed-forward</i>						
OpenScene [110] (Distilled) + Mask3D	7.2	6.2	4.8	10.6	2.6	0.7
Ours (Distilled)	10.2	9.3	7.4	16.5	3.0	0.9

features), OpenMask3D [143], Open3DIS [103] and OpenIns3D [56]. (iii) the feed-forward models that directly map 3D representations to the language embedding space.

Since our approach generates per-Gaussian features whose positions do not align with the original point clouds, we cannot evaluate the 3D instance segmentation results on 3D Gaussians with the ground-truth annotations on point clouds. To address this issue, we utilize the class-agnostic 3D mask proposals to crop out the 3D Gaussians within each mask proposal and recognize the corresponding category. For a fair comparison, we utilize the class-agnostic 3D mask proposals generated by Mask3D [51] for the evaluation of most existing approaches unless otherwise specified. As reported in Table 3.2, our approach achieves the best performance among the per-scene optimization models and feed-forward models in terms of AP₂₅, AP₅₀, and AP. However, we have observed a decline in performance in the tail classes of ScanNet200. The decrease in performance can be attributed to the low-quality reconstruction of smaller objects in the ScanNet200 scene and the misalignment between the original point clouds and the Gaussian points.



Figure 3.3. Qualitative comparison with 2D feature map on ScanNet. PCA visualizations of feature maps obtained via: RADIOv2.5 ViT-H/16 [122], rendered from multi-view fused 3D feature Gaussians and rendered from distilled 3D feature Gaussians.

3.3.3 Evaluation on 3D Question Answering

3D question answering requires a model to generate responses to the natural language queries questioning towards a 3D scene. In this section, we validate our approach on ScanQA [24] for 3D question answering.

Baseline Models. We mainly compare our approach with existing 2D (image/video) and 3D VLMs for different downstream tasks. Existing methods are split into three groups: (i) Discriminative task-specific models perform closed-set classification via detected object proposals. (ii) Zero-shot VLMs include the generalist VLMs trained on internet data while without access to the 3D ground-truth question-answer pairs from several benchmarks. (ii) Fine-tuned VLMs cover the VLMs finetuned with these 3D ground-truth question-answer pairs for particular tasks and multi-tasks.

Results on ScanQA. ScanQA is a subset of ScanNet that contains 41,363 questions about 800 scenes and we evaluate our approach on its validation set which consists of 4,675 questions about 71 scenes. Following prior works, we adopt BLEU scores, METEOR, ROUGE-L,

Table 3.3. Quantitative comparison with SOTA models on ScanQA for 3D QA task. “C” stands for “CIDEr”, “B-4” for “BLEU-4”, “M” for “METEOR”, “R” for “ROUGE”, and “EM@1” for top-1 exact match. Gray indicates evaluation results with refined exact-match protocol as suggested in [53].

	C↑	B-4↑	M↑	R↑	EM@1↑
<i>Task-specific models</i>					
ScanQA [5]	64.9	10.1	13.1	33.3	21.1
3D-VisTA [205]	69.6	10.4	13.9	35.7	22.4
<i>Fine-tuned 3D VLMs</i>					
3D-LLM (FlanT5) [50]	69.4	12.0	14.5	35.7	20.5
LL3DA [18]	76.8	13.5	15.9	37.3	–
Chat-3D v2 [52]	87.6	14.0	–	–	–
LEO [53]	101.4	13.2	20.0	49.2	24.5 (47.6)
Scene-LLM [33]	80	12.0	16.6	40.0	27.2
LLaVA-3D [201]	91.7	14.5	20.7	50.1	27.0 (45.0)
<i>Zero-shot 2D VLMs</i>					
VideoChat2 [74]	49.2	9.6	9.5	28.2	19.2
LLaVA-NeXT-Video [189]	46.2	9.8	9.1	27.8	18.7
Llama3-VILA-1.5 [81]	64.4	–	13.5	35.2	22.4
GPT-4V	59.6	–	13.5	33.4	–
Gemini	68.3	–	11.3	35.4	–
Claude	57.7	–	10.0	29.3	–
Ours (Fused)	72.2	–	16.0	38.8	25.4 (42.1)
Ours (Distilled)	67.5	–	15.2	37.7	23.8 (40.4)

CIDEr and EM (“exact match”) as our evaluation metrics. For existing zero-shot 2D Language-Vision Models (VLMs), we uniformly sample N frames (where $N=8$ in our experiments) as they exclusively take multi-view images as input. As reported in Table 3.3, all evaluated zero-shot 2D VLMs demonstrate relatively poor performance compared to recent 3D VLMs, which are trained with 3D ground-truth annotations and incorporate carefully designed 3D-awareness modules. In contrast, our approach has never been fine-tuned on any 3D QA pairs, and yet outperforms all existing zero-shot VLMs by a large margin and even achieves comparable performance to some fine-tuned 3D VLMs, *i.e.*, LL3DA [18] and 3D-LLM [50]. This significant improvement can be attributed to the more relevant views sampled from the ROI obtained by our 3D Gaussian localization pipeline.

3.3.4 Evaluation on 3D Dense Captioning

3D dense captioning requires the model to localize the objects of interest in a 3D scene and then generate descriptive sentences for these objects. Following the previous work [205, 53],

Table 3.4. Quantitative comparison with SOTA models on Scan2Cap for dense captioning task. “C” stands for “CIDEr”, “B-4” for “BLEU-4”, “M” for “METEOR”, and “R” for “ROUGE”.

	C@0.5↑	B-4@0.5↑	M@0.5↑	R@0.5↑
<i>Task-specific models</i>				
Scan2Cap [24]	39.08	23.32	21.97	44.78
3D-VisTA [205]	61.60	34.10	26.80	55.00
<i>Fine-tuned 3D VLMs</i>				
LL3DA [18]	65.19	36.79	25.97	55.06
LEO [53]	72.4	38.2	27.9	58.1
LLaVA-3D [201]	79.21	41.12	30.21	63.41
<i>Zero-shot 2D VLMs</i>				
LLama-3.2 [29]	3.97	4.81	20.86	28.00
Phi-3 [1]	4.21	4.97	24.36	34.14
InternVL2 [23]	3.68	4.83	23.81	35.23
LLaVA-OneVision [72]	21.29	8.88	25.21	41.44
Owen2-VL [7]	10.96	6.78	25.60	38.93
Llama3-VILA-1.5 [81]	23.34	9.45	26.67	43.71
Ours (Fusion)	30.08	11.74	27.05	46.01
Ours (Distilled)	29.17	11.22	26.58	45.46

we utilize the off-the-shelf segmentation model Mask3D [51] to generate object proposals. Then we replace the object positions obtained from the localization step mentioned in the 3D QA section with these object proposals. To generate the 3D dense captioning of the given object, we follow the same viewpoints sampling strategy described before (Sec 3.3.3) to generate input images for the pre-trained 2D VLM. We utilize the textual instructions that prompt the model to describe the object’s appearance and the spatial relations with nearby objects. As shown in Table 3.4, our approach achieves comparable performance with fine-tune 3D VLMs while consistently surpassing the zero-shot 2D VLMs across all evaluation metrics.

3.3.5 Evaluation on 3D Visual Grounding

3D visual grounding aims to localize the target object in the 3D scene using natural language descriptions. We evaluate the performance on the ScanRefer [14] benchmark. For quantitative comparisons, we found none of the existing zero-shot 2D VLMs can output a reasonable object localization due to the lack of explicit 3D representation. On the other hand, our approach using Gaussian features enables us to calculate the similarity between each Gaussian

Table 3.5. Quantitative comparison with SOTA models on ScanRefer for 3D VG task. We report the top-1 accuracy with 3D bounding box IOU over 25% and 50%.

	Acc@0.25	Acc@0.5
<i>Task-specific models</i>		
ScanRefer [14]	37.3	24.3
ReGround3D [200]	53.1	41.1
<i>Fine-tuned 3D VLMs</i>		
LLM-Grounder [169]	17.1	5.3
3D-LLM [50]	30.3	–
Chat3D-v2 [52]	35.9	30.4
LLaVA-3D [201]	54.1	42.2
<i>Zero-shot 2D VMMs</i>		
None of existing zero-shot 2D LLMs can achieve this task.		
Ours (Fusion)	40.2	16.0
Ours (Distilled)	43.2	18.2

Table 3.6. Ablation study on Rendering loss. We report the 3D instance segmentation results on ScanNet200 and the 3D grounding results on ScanRefer.

Method	ScanNet200			ScanRefer	
	AP ₂₅	AP ₅₀	AP	Acc@0.25	Acc@0.5
Ours w.o. render loss	8.0	7.3	5.8	41.5	17.1
Ours w. render loss	10.2	9.3	7.4	43.2	18.2
improvements	+2.2	+2.0	+2.6	+1.7	+1.1

and the text prompt. We set a threshold to figure out several regions with higher similarity scores that could belong to the target object. As the distribution of 3D Gaussians can be noisy (*e.g.*, like floaters in the air), we further utilize the DBSCAN algorithm to filter out outliers and the final output is a set of 3D bounding boxes that cover each Gaussian grouping. Since this process might result in multiple candidate regions, we choose the one that has the highest IOU score with the ground-truth 3D bounding box when calculating the top-1 accuracy.

3.3.6 Ablation Study

In this section, we conduct several experiments to delve deeper into some details and properties of our proposed approach. **Effectiveness of Rendering Loss.** We first show the effectiveness of the rendering loss used during 3D Gaussian feature distillation. We assess performance on two tasks: 3D instance segmentation on ScanNet200 and 3D visual grounding

Table 3.7. Comparison results with 3DGS-based optimization approaches on ScanNet200. We report the 3D instance segmentation results on ScanNet200 and compared the efficiency with optimization-based approaches.

Model	AP ₂₅	AP ₅₀	AP	Time(min)
Feature-3DGS [195]	17.4	16.0	14.6	~120
LEGaussian [132]	8.2	7.8	7.7	~60
LangSplat [115]	14.3	13.8	12.2	~100
Ours (Fused)	28.9	26.2	22.8	~5
Ours (Distilled)	17.9	17.6	16.3	≤1

Table 3.8. Comparison results with existing 3D VLMs under fine-tuning setting.

Method	ScanQA		Scan2Cap		
	C↑	EM@1↑	B-4@0.5↑	M@0.5↑	R@0.5↑
LEO [53]	101.4	24.5	38.2	27.9	58.1
LLaVA-3D [201]	91.7	27.0	41.1	30.2	63.4
Video-3D LLM [192]	102.1	30.1	41.3	–	–
Ours(Finetuned)	101.9	32.5	30.2	25.9	61.6

on ScanRefer benchmarks. As shown in Table 3.6, incorporating the rendering loss yields improvements of 2.6% AP on ScanNet200 and 1.1% Acc@0.5 on ScanRefer, demonstrating enhanced scene understanding capability.

Comparison with per-scene optimization. To show the quality of distilled Gaussian feature from the feed-forward Gaussian encoder, we compare it with some per-scene optimization approaches, *i.e.*, LangSplat [115], Feature3DGS [195] and LEGaussian [132] on the task of open vocabulary instance segmentation as shown in Table 3.7. Note that the results are evaluated on 12 scenes randomly sampled from the ScanNet validation set due to the computational demands of per-scene optimization methods. Our approach achieves better performance with much less time, *i.e.*, 1 min *v.s.* 100 mins.

Finetune on 3D data. Although the goal of our approach is to enable the 3D spatial understanding ability of 2D VLM without fine-tuning on 3D data, our approach can be further fine-tuned with ground-truth 3D QA pairs similar to LLaVA-3D [201]. The comparison results are listed in Table 3.8. Under a similar fine-tuning setting, our approach achieves comparable results with state-of-the-art on ScanQA and Scan2Cap.

Table 3.9. Ablation study on the number of sampling views. Gray indicates evaluation results using the distillation model.

# Views	C↑	M↑	R↑	EM@1↑
16	68.6 (65.6)	15.5 (15.1)	37.2 (36.5)	23.6 (22.7)
32	71.7 (67.1)	15.8 (15.4)	38.4 (37.4)	25.1 (23.3)
64	72.2 (67.5)	16.0 (15.2)	38.8 (37.7)	25.4 (23.8)

Number of Sampling views. We also show comparison results by varying the number of views sampled from the scene for the evaluation 3D visual question answering. While maintaining a constant input of 8 frames for the 2D VLM, we vary the number of candidate views from which these frames are selected. As presented in Table 3.9, performance tends to improve with an increasing number of candidate sampling views. However, this improvement comes with increased computational cost, presenting a trade-off between performance and efficiency.

3.4 Conclusion

In this work, we presented *SplatDistill*, a novel framework that bridges 3D Gaussian Splatting with foundation models through feature distillation. Given a set of 3D Gaussians, *SplatDistill* predicts the per-Gaussian features that align well with the 2D foundation models via the proposed 3D Gaussian Network. A view-dependent selection and two matching losses are introduced to distill the knowledge of the 2D foundation model onto 3D Gaussians. Based on *SplatDistill*, we design a flexible end-to-end 3D scene understanding pipeline using the extracted Gaussian Features and pre-trained 2D VLMs that achieves improved generalization and efficiency compared to existing methods on various tasks.

Acknowledgments

Chapter 3, in part, is based on the paper from Yang Fu, Sifei Liu, Hongxu Yin, Benjamin Eckart, Jan Kautz, Xiaolong Wang, Arash Vahdat and Chao Liu, “Bridging 2D Vision Language Models to 3D World via Feature Distillation” currently being prepared for submission. The

dissertation author is the primary investigator and author of this paper.

Chapter 4

3D Perception in the Wild — Category-Level 6D Object Pose Estimation

4.1 Introduction

Estimating the 6D object pose is one of the core problems in computer vision and robotics. It predicts the full configurations of rotation, translation and size of a given object, which has wide applications including Virtual Reality (VR) [10], scene understanding [96], and [150, 203, 99, 158]. There are two directions in 6D object pose estimation. One is performing instance-level 6D pose estimation, where a model is trained to estimate the pose of one exact instance with an existing 3D model [43, 109, 73, 163, 104, 20, 42]. However, the learning instance-level model restricts its generalization ability to unseen objects. To achieve generalization in unseen cases, another direction is recently proposed to perform category-level 6D pose estimation using one model [155, 15, 148, 95, 21]. However, the large appearance and shape variance across instances largely increase the difficulty of learning.

To overcome this limitation, Wang *et al.* [155] take the initial step to collect real-world dataset and annotations for category-level 6D pose estimation. Combining synthetic data with free ground-truth annotations, they show the learned model can be generalized to unseen objects within the same category. While this result is encouraging, the generalization ability of the model is still limited by the number and the diversity of the data due to the challenges in annotating 6D object poses. Specifically, only 8,000 images across 13 scenes are collected and annotated from



Figure 4.1. **Left:** we train our model with both synthetic data and real-world data under our proposed semi-supervised setting. **Right:** during inference, given the RGBD images, the object pose can be estimated precisely. Green 3D bounding boxes denote the ground truth, and the red boxes are estimation results via our proposed method

the dataset proposed in [155]. Thus it is still very challenging to generalize 6D pose estimation on diverse objects in complex scenes.

In this paper, we propose to generalize category-level 6D object pose estimation in the wild. To achieve this goal, we introduce a new dataset and a new semi-supervised learning approach. Our key insight is that, while annotating the 6D object pose is challenging, collecting the RGBD videos for these objects without labels is much easier and more affordable. On the other hand, there are infinite ground-truth annotations for synthetic data which come for free. We propose to leverage the benefits from both sides. We first collect a rich object-centric video dataset with diverse backgrounds and object instances using an RGBD camera. We train our model jointly using synthetic data with the free ground-truth 6D pose annotations and the unlabeled real-world RGBD videos via a silhouette matching objective. In this way, our pose estimation model can be generalized to in-the-wild data with minimum human labor. We collect an RGBD video dataset for 6D object pose estimation in the wild, namely *Wild6D*. Each video in the dataset shows multiple views of one or multiple objects (see examples in Fig. 4.1). In total, there are 5166 videos (> 1.1 million images) over 1722 object instances and 5 categories,

which is significantly (300x) larger than the previous 6D object pose estimation dataset [155]. For evaluation, we annotate 486 videos over 162 objects.

Given this dataset, we design a novel semi-supervised model, called **Rendering for Pose estimation Network** (RePoNet). The RePoNet is composed of two branches of networks with a *Pose Network* to estimate the 6D object pose and a *Shape Network* to estimate the 3D object shape. Given an RGBD image of an object, our Pose Network first estimates the Normalized Object Coordinate Space (NOCS) map [155], which will be integrated with lower-layer features to regress the object 9D pose (rotation, translation, and size parameters). Meanwhile, the Shape Network takes a category-level 3D shape prior as an input, and estimates the object shape for the current input instance.

During training, we utilize both the synthetic data with the full ground-truths and the real-world data from Wild6D with foreground segmentation masks (obtained by Mask R-CNN [40]). Given the inputs with synthetic data, we apply the regression losses on both the NOCS map, 6D pose parameters, and object shape. Additionally, given the estimated pose and object shape, we perform differentiable rendering to obtain an object mask projected in 2D. A silhouette matching objective is proposed to compare the difference between the projected mask and the ground-truth mask. With the real RGBD data, although we do not have the 3D ground-truths, we can still learn with the silhouette matching objective by comparing the projected mask against the foreground segmentation. In this way, the gradients are back-propagated through the 6D pose and object shape to adjust both Pose Network and Shape Network for real RGBD data. During inference, we first apply our Pose Network to estimate the NOCS map with a given image. With the NOCS map output, the object pose can be computed by solving the Umeyama algorithm [151]. Some estimation results on Wild6D can be found in Fig. 4.1.

In our experiments, we first evaluate our approach with the dataset proposed in [155]. Our RePoNet shows a improvement over state-of-the-art approaches when trained in a fully-supervised setting (using 3D ground-truths from both real and synthetic data). By using the real-world data without its 3D ground-truth with semi-supervised learning, our performance can

still be on par with previous approaches using full annotations. We then experiment with our Wild6D dataset and consistently show a large performance gain over the baselines on in-the-wild objects.

4.2 Problem Setting

Instance-level 6D Object Pose Estimation. Instance-level 6D pose estimation aims to infer the 3D position and rotation of an object instance, assuming a 3D CAD model for the exact object is provided. We can classify the research in this area into two directions. The first one is to use template matching, which aligns the given CAD model to the observed 3D point clouds [8, 152, 181] or 2D images [76, 141, 57]. Another line of research is to directly regress the 6-DoF pose parameters [163, 153, 28] or predict the keypoints to establish the correspondences to pose using *PnP* or SVD [116, 109, 43, 147, 136, 19]. While these approaches have demonstrated state-of-the-art performance, they are still limited on learning for each individual instance, *i.e.*, YCB [12] or LineMOD objects [47], and cannot generalize to unseen object instance. In this work, we focus on pose estimation at the category level which means our approach can be generalized to unseen objects in the wild.

Category-level 6D Object Pose Estimation. Recently, researchers have proposed to learn a single model for one category of objects instead of just one instance on pose estimation. For example, Wang *et al.* [155] propose a canonical shape representation of different objects called Normalized Object Coordinate Space (NOCS) to handle the instance variations. The object pose and size are calculated by the Umeyama algorithm [151] with predicted NOCS map and the observed points. Follow-up work using the NOCS representation has focused on improving the shape priors [148] and incorporating direct regression of object pose and size [15, 82, 21]. While these approaches can be deployed on unseen object instances, the generalization ability of these models is still limited by the scale and the diversity of real-world annotated data in the REAL275 dataset [155]. Only 13 scenes with 18 real objects in total are presented in REAL275.

Table 4.1. Comparison between Wild6D and existing datasets. Wild6D significantly scale up the number of images, object instances, complexity of the scenes compared to previous RGBD datasets.

Datasets	RGBD	Real	#Categories	#instances	#images
Objectron [4]		✓	9	18K	4M
CAMERA25 [155]	✓		6	184	300K
REAL275 [155]	✓	✓	6	24	8k
<i>Wild6D</i>	✓	✓	5	1.8K	1M

In contrast, we propose a semi-supervised learning approach with RePoNet, which leverages a new large-scale unlabeled dataset Wild6D for training. Both our method and the new dataset are the key components that lead to our goal of generalizing category-level 6D pose estimation in the wild.

4.3 Wild6D Dataset

Existing Datasets. NOCS [155] is the most common dataset for category-level 6D object pose estimation, which consists of the synthetic CAMERA25 dataset and the real-world REAL275 dataset. The synthetic CAMERA25 dataset contains 300,000 RGBD images of 1,085 object instances from 6 categories for training and evaluation. The REAL275 dataset shares the same categories with CAMERA25 but is more challenging under the real-world background. It contains 4,300 RGBD images of 7 scenes for training and 2,750 images of 6 scenes for testing. Due to the limited scale and diversity of real data, the learned models from this dataset cannot generalize to in-the-wild scenarios. Objectron [4] is a large-scale dataset for object pose estimation and tracking. Different from NOCS [155], it consists of a collection of short object-centric video clips captured in the real world. However, no depth maps are provided in Objectron, which might lead to ambiguities in pose estimation.

Wild6D Data Collection. To achieve a categorical 6D pose for real objects, we collect a new large-scale RGBD dataset, named *Wild6D*. Each video in the Wild6D is recorded via the iPhone front camera showing multiple views of objects where RGB images and the corresponding

depth images and point cloud are captured simultaneously. The videos are captured by different turkers with their own iPhones to guarantee the diversity of instances and background scenes. Three videos are taken for each object under different scenes. In total, Wild6D consists of 5,166 videos (>1.1 million images) over 1722 different object instances and 5 categories, *i.e.*, *bottle*, *bowl*, *camera*, *laptop*, and *mug*. Among this data, we split 486 videos of 162 instances to use them as the test set. Table 4.1 summarizes the statistics of Wild6D comparing previous datasets: Wild6D significantly improves the number of images, object instances, and scene complexity.

Wild6D Annotation. To annotate more than 10,000 images in the testing set efficiently, we propose a tracking-based annotation pipeline. Inside a video, we manually annotate the 6D object poses every 50 frames as keyframes. Given the annotation of the keyframe, we implement TEASER++ [168] together with colored ICP [107] to achieve the registration between the keyframe and the following frame and compute the transformation matrix. The ground-truth object pose of the following frame can be obtained by applying the transformation matrix to the keyframe annotation. Following this pipeline, we can obtain accurate ground-truths by only labeling around 5 keyframes per video.

4.4 Method

We propose the **Rendering for Pose estimation network (RePoNet)** using both synthetic data and large-scale unlabeled real-world data in a semi-supervised manner.

RePoNet overview. The RePoNet is composed of two networks including the *Pose Network* to directly estimate the object 6D pose parameters (with the NOCS map as an intermediate representation) and the *Shape Network* to reconstruct the object shape. The outputs from both networks can go through a differentiable rendering [88] module which outputs a segmentation mask.

Semi-supervised learning setting. As shown in Fig. 4.2, we utilize two sets of data for semi-supervised learning: (i) synthetic data with full annotations including NOCS maps, CAD

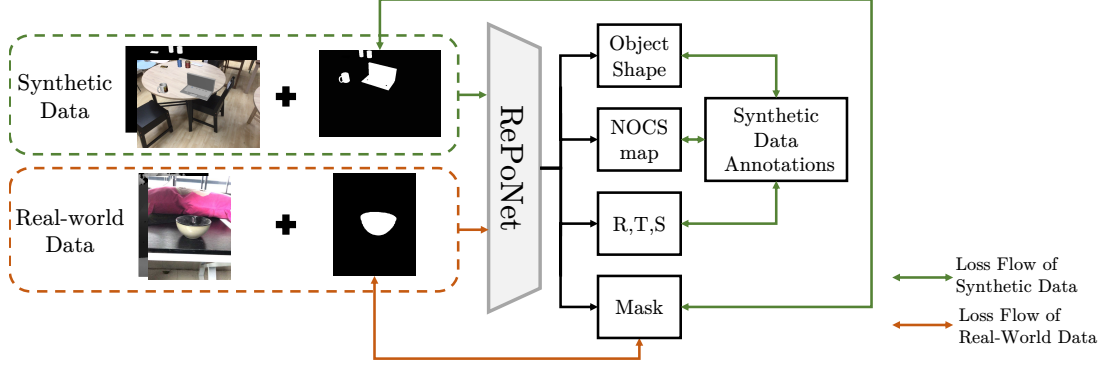


Figure 4.2. Semi-supervised Setting. For the synthetic data, we supervise it with all the annotations. While for the real-world data, we train it by comparing the binary mask generated by the rendering module with the object foreground segmentation.

models, foreground segmentation, and 6D pose parameters, denoted as $\mathbf{D}_{\text{syn}}$; and (ii) a large-scale real-world RGBD dataset Wild6D with estimated foreground masks (using pre-trained Mask R-CNN [40]), denoted as $\mathbf{D}_{\text{real}}$. Both the synthetic data and the real-world data are utilized to train our RePoNet jointly. For the synthetic data, all the ground-truths are used as supervision signals for the *Pose Network* and the *Shape Network*. While for the real-world data, we directly feed the outputs from the *Pose Network* and the *Shape Network* into the differentiable rendering module to generate the binary mask, then train the whole RePoNet by comparing the rendered mask with the object foreground mask (silhouette matching loss). While there can be a more delicate way to deal with the domain gap between simulation and real, we find the direct sharing of parameters during training already provides a simple yet effective solution for generalization.

One important contribution of this framework is to make the whole procedure with RePoNet differentiable, using the Shape Network, and a ConvNet to connect NOCS map to 6D pose parameters in the Pose Network. This allows the gradients to backprop through the network end-to-end. We will explain the architecture and objective details in the following subsections.

4.4.1 Framework Details

Input data pre-processing. Given an RGBD image, we first utilize the off-the-shelf Mask R-CNN [40] model to infer the segmentation mask for each object instance (See supplementary material for segmentation examples). We crop an object of interest from the RGB image and the

corresponding depth map. The pixels in the depth map are back-projected as an observed point cloud. We denote the observation of an object instance as $X \in \mathbb{R}^{H \times W \times 3}$ for RGB channels and $P \in \mathbb{R}^{N_p \times 3}$ for the point clouds, where W, H stands for cropped image resolution and N_p is the number of sampled points.

Feature extraction. We employ the PSPNet [191] to extract the image feature as f_x (Fig. 4.3 green box), and a PointNet [114] to extract the point cloud geometry feature as f_p (Fig. 4.3 blue box). Meanwhile, for each category, we first define a category-level mesh prior $M \in \mathbb{R}^{N_v \times 3}$ to represent objects belong to a specific category, where N_v is the number of vertices of the pre-defined mesh. We also use the PointNet to extract the shape prior feature as f_{cate} (Fig. 4.3 purple box).

Pose Network. Unlike instance-level 6D object pose estimation, where the CAD model of a specific instance is always given as the reference, the Pose Network is expected to work well even without any instance models. To achieve this goal, we leverage the NOCS representation proposed in [155]. The architecture of the Pose Network is shown as the bottom part of Fig. 4.3. The Pose Network takes the RGB features f_x extracted using PSPNet and the point cloud features f_p extracted using PointNet as inputs. The two different modalities are combined as RGBD features using the dense fusion approach proposed in [152]. Specifically, the geometry feature f_p^i of the i th point is concatenated with its corresponding color feature f_x^i and it is fed into a Graph Convolution Network (GCN) [84] (Fig. 4.3 yellow box) to obtain the point-wise RGBD feature, denote as f_{rgbd}^i .

Meanwhile, recall that we utilize a PointNet to obtain the point-wise geometry feature of the category-level shape prior f_{cate} (on the Shape Network branch). The shape prior contains the prior knowledge of the category, *i.e.* coarse canonical shape and pose, but it does not align with the RGBD feature perfectly. To use the information of categorical shape prior and address the misalignment, we first apply a max-pooling operation on point dimension to obtain a global representation, then concatenate it with every point-wise RGBD feature, denoted as f_{nocs}^i , which can be used to estimate the NOCS coordinate. As the size and geometry of real-world object can

be quite diverse, we introduce an **implicit function** $\Phi_{\text{nocs}}(\cdot)$ to deal with objects of the arbitrary shape (Fig. 4.3 gray box with MLP). Specifically, the $\Phi_{\text{nocs}}(\cdot)$ is implemented as a MLP with the point position and the corresponding feature as inputs, and predicts its NOCS coordinate as,

$$\Phi_{\text{nocs}}(f_{\text{nocs}}^i, \mathbf{p}^i \in \mathbf{P}) = \mathbf{p}_{\text{nocs}}^i, \quad \forall i := 1 \dots N_p, \quad (4.1)$$

where $\mathbf{P}$ is the set of input point clouds and N_p is the number of point clouds. The output is the NOCS coordinate $\mathbf{p}_{\text{nocs}}^i$ for each point.

The NOCS map also explicitly reflects the geometric shape information of objects [153]. Thus, the 6D pose, *i.e.*, rotation ($\mathbf{R}$), translation ($\mathbf{T}$), and scale ($\mathbf{S}$) can be inferred from the NOCS map via a simple neural network. In detail, we directly take the RGBD feature f_{rgb}^d and concatenate it with the **intermediate results of NOCS coordinates**. This concatenated feature f_{pose} will be used to predict the object pose $\mathbf{R}, \mathbf{T}$ and $\mathbf{S}$ by a 3-layer ConvNet (Fig. 4.3 gray box with CNN). Using a neural network to connect the NOCS map and the 6D pose parameters makes the whole process differentiable, which is essential for training with the silhouette matching loss. More details of the network architecture can be found in the supplementary materials.

Inference for 6D pose. For 6D object pose estimation, only the NOCS map output of Pose Network is required for inference. The object pose can be computed by solving the Umeyama algorithm [151] with the NOCS map. The 6D pose parameters $\mathbf{R}, \mathbf{T}$ and $\mathbf{S}$ are only used to perform differentiable rendering during training. The reason is that the directly estimated pose parameters from conv module is accurate when fitting the training data, but not as accurate as using NOCS map when generalizing to novel test instances.

Shape Network. The Shape Network (top part of Fig. 4.3) aims to reconstruct the 3D shape of the input object from the given shape prior via mesh deformation. The major usage of this network is for performing differentiable rendering to provide training signals. The Shape Network is not necessary during inference time for 6D object pose estimation.

Given the categorical shape prior representation f_{cate} , the Shape Network borrows the

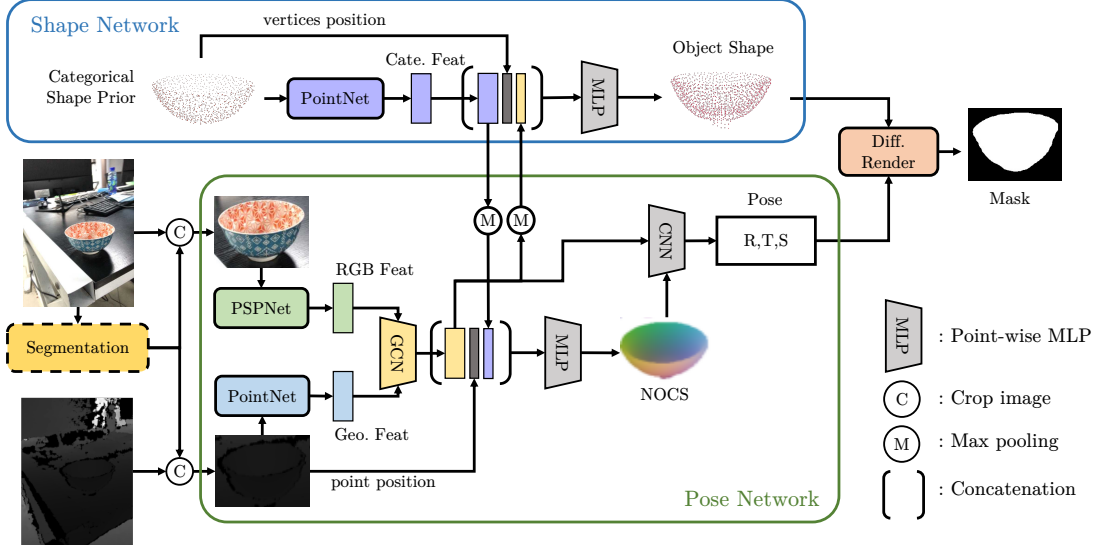


Figure 4.3. Overview of the proposed method. Given the input image and depth map, RePoNet estimates the object pose, NOCS map, and shape simultaneously via Pose Network and Shape Network. These two networks are bridged via the differentiable rendering module. By comparing the predicted binary mask with the input foreground mask, RePoNet can effectively leverage the real-world data without any annotations.

point-wise RGBD feature of the input object $f_{\text{rgb d}}$ from Pose Network and concatenate them together, denoted as f_{shape} . Specifically, we max-pool the features $f_{\text{rgb d}}$ on point dimension and repeat it back with the number of shape prior points for concatenation with f_{cate} . We then predict the per-vertices deformation for the input mesh given f_{shape} . The deformation prediction network $\Phi_{\text{deform}}(\cdot)$ is also an implicit function taking f_{shape} and mesh vertices positions $\mathbf{M}$ as the inputs, and outputs the 3D deformation as,

$$\Phi_{\text{deform}}(f_{\text{shape}}^i, \mathbf{m}^i \in \mathbf{M}) = \mathbf{m}_{\text{delta}}^i, \quad \forall i := 1 \dots N_v, \quad (4.2)$$

where $\mathbf{m}_{\text{delta}}^i$ represents the deformation for each vertex, N_v is the number of the vertices, and $\mathbf{M}_{\text{delta}} = \{\mathbf{m}_{\text{delta}}^i\}_{i=1}^{N_v}$. Then the estimated mesh vertices of the given object is $\mathbf{M}_{\text{deform}} = \mathbf{M} + \mathbf{M}_{\text{delta}}$.

Differentiable rendering. To obtain the supervision from 2D segmentation, we perform differentiable rendering using the outputs from the Shape Network and the Pose Network. Specifically, we feed the estimated object shape $\mathbf{M}_{\text{deform}}$ and estimated object pose $(\mathbf{R}, \mathbf{T}, \mathbf{S})$

into a differentiable rendering module [88] to generate the binary mask of the given object, as shown in Fig. 4.3. Given the rendering output binary mask, we can compare it against the Mask R-CNN segmentations and supervise the training of both pose network and shape network without additional 3D annotations.

4.4.2 Learning Objectives

We define multiple objectives for training the RePoNet, including the loss for 6D pose parameters, NOCS regression loss, shape reconstruction loss using synthetic ground-truths for supervision, and a silhouette matching loss which can be applied to both synthetic data and real data without 6D pose annotations. We introduce each loss as the following.

Disentangled 6D Pose Loss. Inspired by [76, 153, 135], we implement disentangled pose loss via individually supervising the rotation $\mathbf{R}$, translation $\mathbf{T}$ and scale $\mathbf{S}$. Instead of directly computing parametric distances based on rotation matrix, we employ the variant of Point-Matching loss¹ [76, 69]. Additionally, we decouple the translation into the 2D location (o_x, o_y) of the 3D centroid projection on the image plane and the object’s distance to camera t_z . (o_x, o_y) can be approximated as the bounding box center of the given object (c_x, c_y) . Given the camera intrinsics K , the translation can be calculated via Eq. 4.3,

$$\mathbf{T} = K^{-1} t_z [o_x, o_y, 1]^T \quad (4.3)$$

Therefore, given the ground-truth pose annotations $(\mathbf{R}^*, \mathbf{T}^*, \mathbf{S}^*)$, the objective function of 6D pose is formulated as Eq. 4.4,

$$\mathcal{L}_{\text{pose}} = \mathcal{L}_{\mathbf{R}} + \mathcal{L}_{\text{center}} + \mathcal{L}_z + \mathcal{L}_{\mathbf{S}} \quad (4.4)$$

¹We predict the quaternion representation of rotation matrix and follow the strategy in [163, 153] to deal with the symmetry of the object.

Thereby,

$$\mathcal{L}_{\mathbf{R}} = \text{avg}_{\mathbf{x} \in \mathbf{M}_{\text{CAD}}} \|\mathbf{R}\mathbf{x} - \mathbf{R}^*\mathbf{x}\|_2 \quad \mathcal{L}_{\text{center}} = \|o_x - o_x^*, o_y - o_y^*\|_1 \quad \mathcal{L}_z = \|t_z - t_z^*\|_1 \quad \mathcal{L}_{\mathbf{S}} = \|\mathbf{S} - \mathbf{S}^*\|_1 \quad (4.5)$$

where $\mathbf{M}_{\text{CAD}}$ is the ground-truth CAD model of the given object and o_x^*, o_y^*, t_z^* can be computed via Eq. 4.3.

NOCS Regression Loss. To guarantee the pose estimation accuracy, we enforce the predicted NOCS coordinates $\mathbf{P}_{\text{nocs}}$ to be closer to the ground-truth ones $\mathbf{P}_{\text{nocs}}^*$. We use smooth-L1 loss [35],

$$\mathcal{L}_{\text{nocs}} = \sum_{x,y} \begin{cases} 0.5(x-y)^2/\beta, \text{ if } |x-y| < \beta \\ |x-y| - 0.5 * \beta, \text{ otherwise} \end{cases} \quad (4.6)$$

where $x \in \mathbf{P}_{\text{nocs}}^*, y \in \mathbf{P}_{\text{nocs}}$

Reconstruction Loss. Given the ground-truth CAD model $\mathbf{M}_{\text{CAD}}$, the shape estimation is supervised by minimizing Chamfer Distance(CD) between $\mathbf{M}_{\text{deform}}$ and $\mathbf{M}_{\text{CAD}}$ via Eq. 4.7,

$$\mathcal{L}_{\text{recon}} = \sum_{x \in \mathbf{M}_{\text{deform}}} \min_{y \in \mathbf{M}_{\text{CAD}}} \|x - y\|_2^2 + \sum_{y \in \mathbf{M}_{\text{CAD}}} \min_{x \in \mathbf{M}_{\text{deform}}} \|x - y\|_2^2 \quad (4.7)$$

Silhouette Matching Loss. The above three objective functions, $\mathcal{L}_{\text{pose}}$, $\mathcal{L}_{\text{nocs}}$ and $\mathcal{L}_{\text{recon}}$, require the ground-truth information and cannot work with the real-world data without annotations. Thus, we further implement a silhouette matching objective function based on input foreground segmentation and the rendering results. We employ the negative IOU loss [178] to measure the difference between two masks and denote it as $\mathcal{L}_{\text{mask}}$.

Therefore, for the **fully-supervised training** with annotated synthetic data, the total loss function is described in Eq. 4.8, where the λ s are balance parameters.

$$\mathcal{L}_{\text{sup}} = \lambda_1 \mathcal{L}_{\text{pose}} + \lambda_2 \mathcal{L}_{\text{nocs}} + \lambda_3 \mathcal{L}_{\text{recon}} + \lambda_4 \mathcal{L}_{\text{mask}} \quad (4.8)$$

Meanwhile, under the **semi-supervised setting** with both annotated synthetic data and unlabeled data, the total loss function is then formulated as:

$$\mathcal{L}_{\text{semi}} = \mathbb{1}_{d \in \mathbf{D}_{\text{syn}}} (\lambda_1 \mathcal{L}_{\text{pose}} + \lambda_2 \mathcal{L}_{\text{nocs}} + \lambda_3 \mathcal{L}_{\text{recon}}) + \lambda_4 \mathcal{L}_{\text{mask}} \quad (4.9)$$

where $\mathbb{1}_{d \in \mathbf{D}_{\text{syn}}}$ is the indicator function.

4.5 Experiments

4.5.1 Implementation Details

Semi-supervised setting. We use the training data of CAMERA25 [155] along with the corresponding annotations and jointly train the model with images of REAL275 [155] or Wild6D without any 6D pose annotations. After cropping the object from the RGBD image, we first resize it to 192×192 and then randomly sample 1,024 points from both color image and depth map. To obtain the categorical shape prior, we choose a CAD model per category from the CAMERA25 training set manually and reduce its number of vertices to 1,024 as well. More configurations of RePoNet have been specified in supplementary materials. We adopt Adam [66] to optimize our model with the initial learning rate of 0.0001. The learning rate is halved every 10 epochs until convergence. We empirically set the balance parameters $\lambda_1, \lambda_2, \lambda_3$ and λ_4 to 0.2, 2.0, 5.0 and 0.2, respectively. **Fully-supervised setting.** We share the same architecture and configurations with semi-supervised training but utilize all annotations of CAMERA25 [155] and REAL275 [155].

4.5.2 Ablation Study

Implicit function and NOCS map. We first validate the effectiveness of two components in our approach: implicit function and intermediate NOCS map as described in previous method section. We evaluate the performance under fully-supervised setting and semi-supervised setting on REAL275 [155] and report the results in Table 4.2. Under the fully-supervised setting,

Table 4.2. Ablation on Implicit function and intermediate NOCS map. We evaluate on REAL275 and the best results are highlighted in bold. w./w.o the nocs map refers to whether using the NOCS map as an intermediate result for pose regression. w./w.o. the implicit function stands for whether including the point/mesh coordinates in NOCS regression and shape reconstruction.

NOCS map	Implicit function	IOU _{0.5}	IOU _{0.75}	5 degree 5cm	10 degree 5cm	IOU _{0.5}	IOU _{0.75}	5 degree 5cm	10 degree 5cm
		Fully supervised				Semi-supervised			
		77.0	55.7	35.4	63.2	71.7	47.9	22.1	47.5
✓		79.1	59.1	36.9	65.3	74.6	49.9	29.7	57.1
	✓	79.2	60.0	39.8	66.6	72.2	47.7	24.3	51.6
✓	✓	81.1	63.7	40.4	68.8	76.0	52.2	33.9	63.0

Table 4.3. Ablation study on pose regression loss and silhouette matching loss. We evaluate on REAL275 and the best results are highlighted in bold

silhouette loss	pose loss	IOU _{0.5}	IOU _{0.75}	5 degree 5cm	10 degree 5cm	IOU _{0.5}	IOU _{0.75}	5 degree 5cm	10 degree 5cm
		Fully supervised				Semi-supervised			
✓		78.1	61.9	39.7	65.2	74.8	47.1	27.9	59.7
	✓	76.7	58.2	30.7	59.7	73.1	46.0	18.7	47.9
✓	✓	81.1	63.7	40.4	68.8	76.0	52.2	33.9	63.0

by using the implicit function, the performance on IOU_{0.5} and 5 degree, 5cm is improved by 2.2% and 4.4%, respectively. A similar improvement can also be observed for the model with an intermediate NOCS map. The best performance is achieved when combining the implicit function and NOCS map together. And the same conclusion can also be obtained under the semi-supervised setting as shown in Table 4.2.

Pose Loss and Silhouette Matching Loss. Besides the network architecture, we also show the effectiveness of the disentangled 6D pose loss and silhouette matching loss. The performance of the model w./w.o. the pose regression loss and w./w.o. the silhouette matching loss are listed in Table 4.3. Clearly, both pose loss and silhouette matching loss can enhance the object pose estimation performance under both fully-supervised and semi-supervised settings. For instance, the silhouette matching loss improves the IOU_{0.75} and 10 degree 5cm by 5.5% and 9.1% compared with the model without it under the fully-supervised setting. This significant improvement shows the rendering module can effectively encourage communication between two networks, thereby boosting the pose estimation performance.

Table 4.4. Ablation study of proposed semi-supervised method with different training data. We evaluate the performance on REAL275 test set. “RePoNet-semi” and “RePoNet-sup” represent for the model trained w./w.o the 3D annotations of NOCS REAL275 respectively. Here we conduct experiments on five categories sharing by both REAL275 and Wild6D, *i.e.* bottle, bowl, mug, laptop and camera.

Methods	Training Data			IOU _{0.25}	IOU _{0.5}	IOU _{0.75}	5 degree 2cm	5 degree 5cm	10 degree 2cm	10 degree 5cm
	CAMERA25	REAL275	Wild3D							
RePoNet-semi	✓			83.8	73.5	45.7	18.2	21.8	35.6	47.9
	✓		✓	85.0	74.3	52.5	23.6	26.1	45.2	53.8
	✓	✓		85.8	76.9	49.2	29.1	31.3	48.5	56.8
	✓	✓	✓	86.0	78.5	51.8	30.8	36.0	48.7	59.7
RePoNet-sup	✓	✓		86.6	83.0	63.2	31.1	37.0	51.6	64.6

Leveraging unlabeled real data. Additionally, we report the performance of the proposed semi-supervised approach with training data from different sources in Table 4.4. As the baseline model, we first train it only with the synthetic data from CAMERA25 in a fully-supervised manner. Then, by using the unlabeled Wild3D RGBD videos, the performance is significantly improved by 6.8% and 5.4% on IOU_{0.75} and 5 degree, 2cm. Similarly, the performance gains can be observed when adopting the REAL275 data without its annotations. Furthermore, with both REAL275 and Wild3D data, our model outperforms the baseline model by a large margin and approaches the fully-supervised model trained on REAL275 which can be served as the upper bound.

4.5.3 Comparison with State-of-the-art Methods

Performance on REAL275. We split all existing methods into two groups by whether using full annotations of real data and compare their performance with RePoNet under both fully-supervised setting and semi-supervised setting, denoted as “RePoNet-sup” and “RePoNet-semi” respectively. As listed in Table 4.5, RePoNet-sup can achieve competitive results among all existing approaches under the fully-supervised setting. Note our method is complimentary to the techniques proposed in SGPA, and our key contribution and focus lies on the semi-supervised counterpart for generalization. As shown in the bottom part of Table 4.5, the RePoNet-semi

Table 4.5. Comparison of our approach with the SOTA methods on REAL275. Note that all SOTA methods listed in the top part are trained with full annotations of NOCS dataset including CAMERA25 and REAL275, while the bottom part is methods without using any real annotations. The best results are highlighted in bold.

Methods	IOU _{0.5}	5 degree 2cm	5 degree 5cm	10 degree 5cm
NOCS [155]	78.0	7.2	10.0	25.2
CASS [15]	77.7	–	23.5	58.0
Shape-Prior [148]	77.3	19.3	21.4	54.1
FS-Net [21]	92.2	–	28.2	60.8
DualPoseNet [82]	79.8	29.3	35.9	66.8
SGPA [16]	80.1	35.9	39.6	70.7
RePoNet-sup	81.1	35.1	40.4	68.8
CPS++ [95]	17.7	–	–	≤ 22.3
RePoNet-semi	76.0	30.7	33.9	63.0

Table 4.6. Comparison of our approach with the SOTA methods on Wild6D. The best results are highlighted in bold.

Methods	IOU _{0.5}	5 degree 2cm	5 degree 5cm	10 degree 5cm
CASS [15]	1.04	0	0	0
Shape-Prior [148]	32.5	2.6	3.5	13.9
DualPoseNet [82]	70.0	17.8	22.8	36.5
RePoNet-syn	66.7	26.0	30.8	40.3
RePoNet-semi	70.3	29.5	34.4	42.5

outperforms CPS++ [95] by a large margin under the semi-supervised setting ². Moreover, compared with the existing fully-supervised methods, RePoNet-semi achieves a comparable or even better performance without any 6D pose annotations.

Performance on Wild6D. Finally, we evaluate the performance of RePoNet on the Wild6D testing set as reported in Table 4.6. For some existing works, we directly test their pre-trained models trained on CAMERA75 along with REAL275. Since FS-Net [21] does not release the model, we cannot experiment on it. It can be observed that the pre-trained models cannot generalize to Wild6D due to the limited diversity of real data during training. For example, the performance of Shape-Prior [148] on three pose estimation metrics are all lower than 15%. On

²In Table 4.5, CPS++ only reports the performance under 10 degree, 10 cm.

the other hand, the RePoNet with semi-supervised learning on Wild6D achieves 29.5% and 34.4% on 5 degree, 2cm and 5 degree, 5cm, which is $10\times$ better than Shape-Prior [148]. This significant improvement shows the better generalization ability of RePoNet. Also, the superior performance of RePoNet-semi over RePoNet-syn shows our proposed semi-supervised training can effectively leverage the in-the-wild data.

4.6 Conclusion

In this work, we consider the problem of generalizing 6D pose estimation to in-the-wild objects. Most of existing work are restricted in constrained environments due to limited number of annotated data. In an effort to resolve this limitation, we collect *Wild6D*, a new RGBD dataset with diverse instances and backgrounds. Instead of annotating them, we also propose RePoNet, a network that can leverage both the synthetic data and real-world RGBD data. Without using any 3D annotations on real-world data, RePoNet outperforms state-of-the-art methods on existing datasets and Wild6D test set by a large margin.

Limitations. Our proposed RePoNet may be hard to generalize to unseen categories. This is because RePoNet depends on the categorical mesh prior which is pre-defined for each category leading to the learnt deformation model fixed to a specific category. Alternative way to address this problem is to represent object shape via a deformation of a unit sphere to get rid of the category-level prior.

Acknowledgments

Chapter 4, in part, is based on the paper from Yang Fu and Xiaolong Wang, “Category-Level 6D Object Pose Estimation in the Wild: A Semi-Supervised Learning Approach and A New Dataset” published in Advances in Neural Information Processing Systems (NeurIPS), 2022. The dissertation author is the primary investigator and author of this paper.

Chapter 5

3D Perception in the Wild — Scene Reconstruction as Mapping Priors for Autonomous Driving

5.1 Introduction

Precisely recognizing and localizing objects in 3D world, also known as 3D detection, is crucial for the safety-critical L4+ autonomous driving. Due to the highest standard quality bar, it is challenging despite multiple sensors (LiDAR, camera, and radar) has been employed. However, sensors are not robust to low-visibility and low-fidelity scenarios, such as occlusions, long-range sparsity, and sensor noise due to extreme weather and lighting conditions. As described in Fig. 5.1, vehicles that are difficult to distinguish using only LiDAR data. Therefore, previous studies [164, 167] have shown that leveraging High-Definition (HD) maps can improve 3D detection quality.

High-Definition (HD) maps, in particular, encode detailed static and semantic information, such as road layouts, lane markings, and infrastructure locations, are commonly used by L4 autonomous driving vehicles today. Several works [165, 87, 77, 79, 164, 167] have demonstrated how HD maps can be leverage to mitigate ambiguities and noisy caused by sparsity and occlusion of the sensor data. However, building HD maps is not scalable as it relies heavily on manual human annotation of every road feature, making it costly to create and challenging to scale [17,

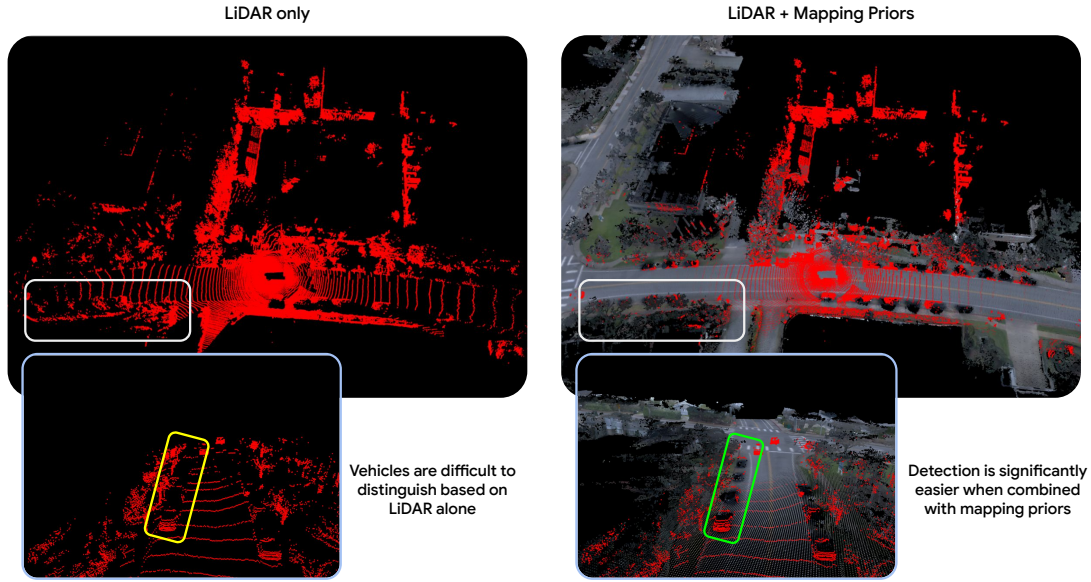


Figure 5.1. Scene Reconstruction as Mapping Priors for 3D Detection. (Left) Standard 3D detection using only sparse LiDAR data struggles to identify and distinguish multiple vehicles (yellow boxes) amidst background noise. (Right) Our method fuses the sparse LiDAR with dense, reconstructed mapping priors, providing rich context that resolves ambiguities (green boxes) and achieves accurate 3D detection.

106, 159].

Instead of using non-scalable HD Maps, in this paper, we propose to leverage scalable reconstruction methods [61, 63, 98, 144, 172, 64] as mapping priors to enhance 3D detection. Reconstruction methods can create dense maps with photorealistic appearance and geometric information, providing a scalable and automated way to generate mapping priors directly from collected sensor data. We investigate two such priors: surfels [111] and 3D Gaussians (3DGS) [61], which can be automatically generated from camera and LiDAR data collected during single vehicle pass or multiple traversal data. Surfel representation is simple, computationally efficient, and preserves rich geometric and appearance information. However, its primary limitation stems from a heavy reliance on LiDAR points, which can result in noisy or incomplete maps in areas of sparse data. On the other hand, 3DGS provides a more robust and comprehensive representation with the extra compute cost compared to surfel reconstruction. It is initialized with LiDAR data and optimized using camera data, enabling it to potentially

recover regions where LiDAR data is sparse or missing. As demonstrated in Fig 5.1, with the mapping priors as the background, vehicles that are difficult to distinguish using only LiDAR data become significantly easier to detect.

To harness the complementary advantages of these priors and the raw sensor data, we employ a unified and flexible framework to integrate surfels, 3DGS, LiDAR, and camera data to leverage the complementary advantages of both mapping priors. In particular, we adopt modality-dependent encoders and dynamic voxelization [139] for feature extraction. To fuse the mapping prior information, we propose a gated fusion module to effectively modulate the influence of mapping priors while preserving the reliable LiDAR features. Moreover, we introduce a mixed-modalities training strategy to handle arbitrary combinations of these modalities at both training and inference time.

5.2 Method

In this section, we describe MPA3D, Mapping Priors Augmented 3D object detection. The overall framework is shown in Figure 5.2. Built upon [139], we take two types of mapping priors, *e.g.*, surfels and 3DGS, as additional inputs and employ a gated fusion strategy to integrate them with LiDAR and camera data. In the following subsections, we first introduce a scalable data pipeline to generate huge amount of surfels and 3DGS without any human annotations. We then demonstrate how to integrate multi-sensor data with mapping priors.

5.2.1 Scalable Mapping Prior Generation

Surfel Map Reconstruction. Following [172], we generate surfel map with faithful preservation of sensor information while maintaining low computational and storage costs. Specifically, we discretize the scene into a 3D voxel grid with fixed voxel size of 0.25m and process multi-traversal LiDAR scan data. For each voxel, we construct a surfel disk [111] by estimating the mean coordinate, surface normal, and mean color from its binned LiDAR points and corresponding camera pixels. As a result, we represent a scene with a set of surfel disks:

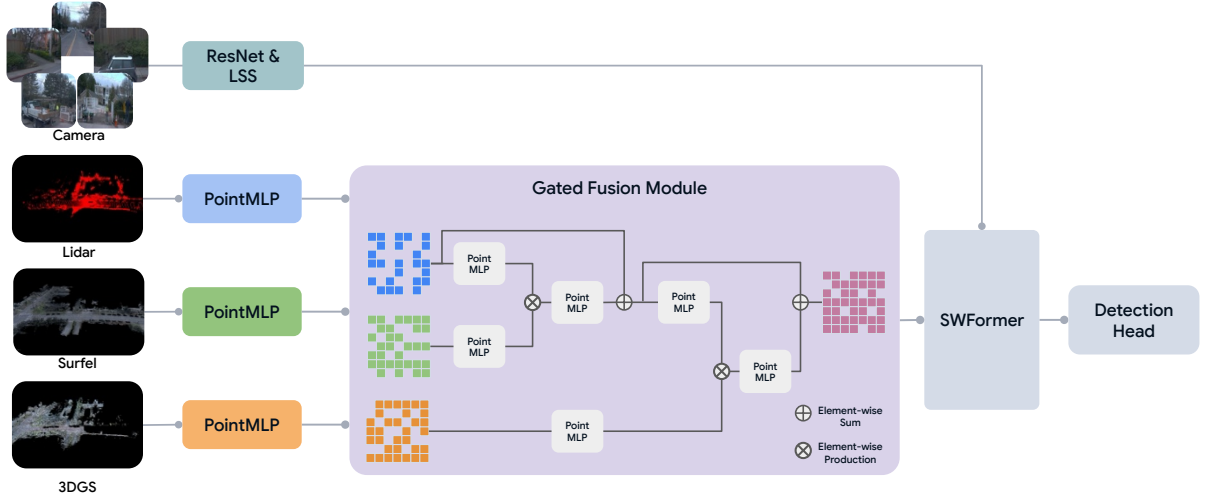


Figure 5.2. Overview of *MPA3D*. The framework processes multi-view camera images via a ResNet backbone and Lift-Splat-Shoot [112] to generate BEV representations. Simultaneously, LiDAR point clouds and mapping priors (e.g., surfels and 3DGS) are encoded using PointMLPs. A novel Gated Fusion Module adaptively integrates these sparse 3D modalities through hierarchical, element-wise gating operations. Finally, the fused geometric features are combined with the image representations and processed by an SWFormer backbone and a detection head to yield 3D bounding box predictions.

$\mathcal{S} = \{x_i, n_i, c_i\}_{i=1:N_S}$, where x , n and c refer to the surfel position, normal and color, respectively and N_S is the number of surfel disk. Since the creation of each surfel is relatively independent, we can partition a large region into multiple sub-regions and conduct reconstruction with massive parallelism, which allows us easily scale up surfel map creation to multiple cities.

3DGS Map Reconstruction. Surfel maps are heavily reliant on LiDAR data, making them susceptible to noise and artifacts from sensor perturbations. To address this, we propose to use 3DGS as a complementary mapping prior. Specifically, a 3DGS scene is represented by a set of 3D Gaussians:

$$\mathcal{G} = \{(\boldsymbol{\mu}, \text{SH}, \boldsymbol{r}, s, \alpha)_i\}_{i=1:N_G}$$

where each Gaussian i is defined by its 3D position $\boldsymbol{\mu}$, spherical harmonics (SH) coefficients SH, quaternion rotation $\boldsymbol{r}$, scale s , opacity α , and N_G is the total number of Gaussians. We initialize the Gaussian positions $\boldsymbol{\mu}$ from LiDAR data, but subsequently optimize all attributes (including

position) by minimizing a photometric loss against the camera images. This is a key difference from surfel maps, whose disk positions are fixed by the initial LiDAR points. By refining the Gaussian positions during optimization, 3DGS can effectively correct for noise and densify regions with sparse or incomplete LiDAR data. Note that we implemented custom CUDA kernels to perform 3D Gaussian ray-tracing (similar to [161]) to better represent fine geometry while still maintain compute efficiency, instead of the original splatting implementation [61].

Dynamic Object Removal. Since 3DGS reconstruction assumes a static scene, we must first remove dynamic objects. For the training set, we utilize the 3D bounding box annotations to remove dynamic points during initialization and project object-specific LiDAR points into 2D masks. These masks are then used to exclude moving objects during 3DGS optimization. For inference, we first run our model in its map-free configuration (enabled by our mix-training strategy) to predict initial 3D bounding boxes. These predictions are then used to create the necessary masks for on-the-fly map generation. We leave the accurate dynamic reconstruction to future work.

5.2.2 Mapping Prior Augmented 3D Detection

Our model architecture, illustrated in Fig. 5.2, processes camera data I , LiDAR data P , and mapping priors M . First, modality-dependent encoders extract features, which are then projected into a shared Bird’s-Eye-View (BEV) space, similar to [91]. These BEV features are subsequently processed by our proposed gated fusion module, which leverages the mapping priors to enhance the sensor-derived features. Finally, the fused BEV representation is fed into a sparse-window transformer [139] and a detection head to produce the final 3D object detections.

Modality-Dependent Feature Extraction. We employ different encoders tailored to each input modality after transforming it to the same vehicle coordinate. For LiDAR points, we follow [139] to extract point-level features into the BEV space: raw points are first grouped into voxels via dynamic voxelization [198], then processed through PointMLP [94] to generate point-wise features, which are subsequently aggregated into BEV features. For camera data,

we use the lift-splat-shoot (LSS) [112] module to lift image features into 3D frustums and splat them into the shared BEV space. For surfel points, we treat surfel disk centers as pseudo-LiDAR points and apply the similar encoder architecture as LiDAR encoder, leveraging their geometric properties. For 3DGS, we convert each Gaussian into a point-like representation characterized by all Gaussian attributes. Notably, we only utilize the first order of SH coefficient and adopt the 6D continuous rotation representation [197] instead of quaternions to encode colors and orientations. We introduce an additional projection layer for each modalities to align them into the same embedding space. As a result, we have a set of LiDAR, camera, surfel and 3DGS features denoted as $\{f_{\text{lidar}} \in \mathbb{R}^{N_{\text{lidar}} \times d}, f_{\text{camera}} \in \mathbb{R}^{N_{\text{camera}} \times d}, f_{\text{surfel}} \in \mathbb{R}^{N_{\text{surfel}} \times d}, f_{\text{Gaussian}} \in \mathbb{R}^{N_{\text{Gaussian}} \times d}\}$, where N_* refers to the number of voxels after dynamic voxelization.

Feature Enhancement via Gated Fusion. Since we employ the same dynamic voxelization under the same vehicle coordinate for LiDAR, surfel and 3DGS inputs, the resulting lidar feature f_{lidar} , surfel feature f_{surfel} and Gaussian feature f_{gaussian} share the same BEV grid structure with $N_{\text{lidar}} = N_{\text{surfel}} = N_{\text{gaussian}}$. A straightforward baseline is to average features from different modalities that occupy the same voxel using a segment mean operation over the sparse feature tensor. However, this naive averaging approach has a critical limitation: it is biased toward the modality with the highest point density. For instance, if a voxel contains 95 LiDAR points and only 5 Gaussians, the averaged feature is dominated by LiDAR (95%), effectively marginalizing the complementary information from Gaussians. This density imbalance prevents effective multi-modal fusion and fails to leverage the unique strengths of each representation.

To mitigate this issue, we propose a gated fusion module to learn adaptive weighting based on local scene characteristics. As described in Fig. 5.2, we first aggregate features within a single voxel from different modalities independently using segment mean operation, yielding modality-level features $\bar{f}_{\text{lidar}}$, $\bar{f}_{\text{surfel}}$, and $\bar{f}_{\text{gaussian}}$. We treat the LiDAR feature $\bar{f}_{\text{lidar}}$ as the primary feature, which acts as a gate for the mapping priors. We employ several PointMLPs for internal feature transformations and for output projections, denoted as $\sigma_* \in \mathbb{R}^\times$ and $\phi_* \in \mathbb{R}^{d \times d}$. First, we

compute the gated contribution from the surfel map:

$$\alpha_{\text{surfel}} = \text{Swish}(\sigma_{\text{in}}(\bar{f}_{\text{lidar}})) \cdot \sigma_{\text{surfel}}(\bar{f}_{\text{surfel}})$$

Here, α_{surfel} represents the modulated feature contribution of the surfel map, informed by the LiDAR context and $\text{Swish}(\cdot)$ stands for Swish activation function [44]. The intermediate fused feature is then computed via a residual connection:

$$f_{\text{inter}} = \phi_{\text{surfel}}(\alpha_{\text{surfel}}) + \bar{f}_{\text{lidar}}$$

We then repeat this gated process for the 3D Gaussian prior, using the intermediate fused feature as the new gate:

$$\alpha_{\text{Gaussian}} = \text{Swish}(\sigma_{\text{inter}}(f_{\text{inter}})) \cdot \sigma_{\text{Gaussian}}(\bar{f}_{\text{Gaussian}})$$

The final fused feature f_{fused} is obtained by combining this new contribution with the intermediate feature:

$$f_{\text{fused}} = \phi_{\text{Gaussian}}(\alpha_{\text{Gaussian}}) + f_{\text{inter}}$$

This design allows the network to adaptively modulate the influence of mapping priors while preserving the reliable LiDAR features through the skip connection. Finally, f_{fused} is concatenated with the dense camera feature f_{camera} through $f_{\text{final}} = [f_{\text{camera}}, f_{\text{fused}}] \in \mathbb{R}^{(N_{\text{camera}} + N_{\text{lidar}}) \times d}$. This combined BEV representation serves as the input to the subsequent sparse-window transformer.

5.2.3 Mixed-Modality Training Strategy

As previously discussed, our pipeline can effectively integrate and fuse mapping priors with common sensor data. However, in practice, the availability of these mapping priors cannot be guaranteed: Surfel maps and 3D Gaussian maps may be unavailable due to reconstruction failures caused by insufficient data coverage or challenging environmental conditions. Moreover, during inference, we assume no prior knowledge about 3D bounding boxes is available. Therefore,

our model must robustly handle arbitrary combinations of input modalities without relying on complete mapping priors or pre-existing annotations.

To address this challenge, we propose a mix-training strategy that enables the model to operate effectively under varying modality availability. During training, we randomly sample different combinations of input modalities for each training sample, simulating diverse real-world scenarios. In particular, we assume the sensor data, *e.g.*, camera and LiDAR, always exist and define the following modality combinations: sensor + surfel, sensor + 3DGS, sensor + surfel + 3DGS. For each training batch, we randomly drop surfel or 3DGS with predefined probabilities. When a modality is dropped, its corresponding feature is set to zero, and the gated fusion module learns to automatically suppress its contribution through the attention weights. The gated fusion module enables this flexibility through three key design choices. First, the modality-independent aggregation allows missing modalities to contribute zero features without architectural changes. Second, the learnable attention weights automatically suppress contributions from absent or unreliable modalities; for example, the network learns to set $\alpha_{\text{surfel}} \approx 0$ when $\tilde{f}_{\text{surfel}} = \mathbf{0}$. Third, the residual connection with the LiDAR feature $\tilde{f}_{\text{lidar}}$ ensures stable performance. If all mapping priors are absent, the output f_{fused} naturally defaults to $\tilde{f}_{\text{lidar}}$. This design handles any modality combination without requiring explicit masking or conditional branches. This mix-training strategy forces the network to learn robust feature representations that do not over-rely on any single modality. During inference, the model seamlessly adapts to whatever modalities are available without requiring architectural modifications or retraining.

5.2.4 Training Objectives

We follow SWFormer [139] to regress 3D bounding boxes using a combination of heatmap, bounding box, and foreground segmentation losses. Specifically, the heatmap loss L_{hm}^c is computed as a penalty-reduced focal loss [196, 83] for each object class c . For 3D bounding box regression, we parameterize boxes as $\mathbf{b} = \{d_x, d_y, d_z, l, w, h, \theta\}$, where $\{d_x, d_y, d_z\}$ are center offsets relative to voxel centers, and $\{l, w, h, \theta\}$ represent dimensions and heading.

Following [140], the bounding box loss L_{bbox}^c comprises a bin loss for heading, a Smooth L1 loss for other parameters, and an IoU loss [193] to improve overall accuracy. Additionally, we jointly train a class-aware foreground segmentation task, where each voxel is supervised by a binary focal loss L_{seg}^c for each class. The total loss is defined as:

$$L = \sum_c (\lambda_{\text{hm}} L_{\text{hm}}^c + \lambda_{\text{bbox}} L_{\text{bbox}}^c + \lambda_{\text{seg}} L_{\text{seg}}^c)$$

where the loss weights are set to $\lambda_{\text{hm}} = 1.0$, $\lambda_{\text{bbox}} = 2.0$, and $\lambda_{\text{seg}} = 1.0$ during training.

5.3 Experiments

5.3.1 Datasets and Metrics

We conduct experiments and ablation studies mainly on the Waymo Open Dataset (WOD) [138]. The dataset contains 1,150 scenes, split into 798 training scenes, 202 validation scenes, and 150 test scenes. Each scene contains approximately 200 frames, where each frame captures a full 360-degree view around the ego-vehicle. We produce surfel map and 3DGS map for each scene based on LiDAR and camera data from WOD or in-house sensor data. Additionally, we collect around 7M additional sequences for pretraining, and 600k of them have map prior data. These additional sequences enable us to pre-train the mapping prior encoders on large-scale diverse scenarios, improving their feature extraction capabilities before fine-tuning on WOD.

The evaluation metrics include mean average precision (mAP) and mAP weighted by heading accuracy (mAPH), computed at 3D intersection-over-union (IoU) thresholds of 0.5, 0.5 and 0.7 for pedestrian, cyclist and vehicle, respectively. The metrics are further broken down into two difficulty levels: LEVEL_1 (L1) for objects with more than five LiDAR points and are not marked as “hard”, and LEVEL_2 (L2) for objects with at least one LiDAR point. Following common practice [139], the detection range is limited to 75 meters.

5.3.2 Implementation Details

Model Architecture and Training. During training, we ignore all ground truth boxes with fewer than five points inside. We set the voxel size of dynamic voxelization to 0.2m and the maximum voxel number to 250K. The SWFormer architecture consists of five transformer blocks with [4, 6, 4, 6, 4] layers respectively. Each transformer layer has 256 channels, 8 attention heads, and an MLP expansion ratio of 2. The model is trained end-to-end using 256 TPU cores with the LAMB optimizer [177] for 20K steps. We use a cosine learning rate schedule starting from $1e^{-5}$ and decaying to zero until the end of training.

Data augmentation. We adopt several 3D data augmentation techniques described in [71]. In particular, we apply random rotation with yaw angle uniformly sampled from $[-\pi, \pi]$ with probability of 0.74, random flipping along y-axis with probability of 0.5, random scaling with scaling factor uniformly chosen within $[0.95, 1.05)$ and random point dropout with probability of 0.05. To ensure the same augmentations are consistently applied to the camera data, instead of transforming the camera extrinsics and intrinsics, we directly transform the points with camera features lifted from Lift-Splat-Shoot (LSS) [112]. For mapping priors (surfel and 3DGS), we apply the same spatial transformations to their coordinates and orientations to preserve alignment across all modalities.

Efficiency of scalable mapping prior generation. To optimize mapping prior generation throughput, we implement scalable pipelines for both surfel and 3DGS using the Apache Beam API in a MapReduce framework. Using this framework, we can generate both surfel and 3DGS maps for 600,000 scenes within 10 days with massive parallelism, using thousands of CPU cores. This automated generation process represents an affordable computational budget compared to manual annotation of HD maps, which typically requires extensive human labor.

Table 5.1. Performance comparisons on WOD validation set. The metrics are 3D AP and APH for both L1 and L2 difficulties. We use **bold numbers** to highlight the best results.

Method	Overall L1		Overall L2		Vehicle L1		Vehicle L2		Pedestrian L1		Pedestrian L2		Cyclist L1		Cyclist L2	
	AP	APH	AP	APH	AP	APH	AP	APH	AP	APH	AP	APH	AP	APH	AP	APH
SECOND [166]	67.2	63.1	61.0	57.2	72.3	71.7	63.9	63.3	68.7	58.2	60.7	51.3	60.6	59.3	58.3	57.0
PointPillar [70]	69.0	63.5	62.8	57.8	72.1	71.5	63.6	63.1	70.6	56.7	62.8	50.3	64.4	62.3	61.9	59.9
Part-A2-Net [134]	73.6	70.3	66.9	63.8	77.1	76.5	68.5	68.0	75.2	66.9	66.2	58.6	68.6	67.4	66.1	64.9
SST [30]	74.5	71.0	67.8	64.6	74.2	73.8	65.5	65.1	78.7	69.6	70.0	61.7	70.7	69.6	68.0	66.9
Centerpoint [176]	76.1	73.5	70.0	67.6	75.7	75.2	67.9	67.4	77.6	71.6	70.1	64.4	74.9	73.8	72.1	71.0
CenterFormer [199]	75.6	73.2	71.4	69.1	75.0	74.4	69.9	69.4	78.0	72.4	73.1	67.7	73.8	72.7	71.3	70.2
PillarNet-34 [131]	77.3	74.6	71.0	68.5	79.1	78.6	70.9	70.5	80.6	74.0	72.3	66.2	72.3	71.2	69.7	68.7
PV-RCNN++ [133]	78.1	75.9	71.7	69.5	79.3	78.8	70.6	70.2	81.3	76.3	73.2	68.0	73.7	72.7	71.2	70.2
DSVT-Voxel [154]	80.3	78.2	74.0	72.1	79.7	79.3	71.4	71.0	83.7	78.9	76.1	71.5	77.5	76.5	74.6	73.7
SWFormer [139]	-	-	-	-	77.8	77.3	69.2	68.8	80.9	72.7	72.5	64.9	-	-	-	-
FSDv1 [31]	79.6	77.4	72.9	70.8	79.2	78.8	70.5	70.1	82.6	77.3	73.9	69.1	77.1	76.0	74.4	73.3
FSDv2 [32]	81.8	79.5	75.6	73.5	79.8	79.3	71.4	71.0	84.8	79.7	77.4	72.5	80.7	79.6	77.9	76.8
HEDNet [184]	81.6	79.7	75.6	73.7	80.9	80.5	73.1	72.7	84.6	80.2	77.1	72.8	79.4	78.5	76.6	75.6
SAFDNet [185]	81.7	79.7	75.5	73.6	80.5	80.0	72.5	72.1	84.7	80.2	77.1	72.9	79.8	78.8	76.9	75.9
Centerpoint 2f [176]	77.5	75.8	71.7	70.1	76.4	75.9	68.7	68.2	79.2	75.6	71.9	68.5	76.8	75.9	74.4	73.5
HEDNet 4f [184]	83.6	82.3	78.1	76.8	82.4	81.9	75.1	74.6	86.3	83.6	79.4	76.8	82.2	81.4	79.9	79.1
SAFDNet 4f [185]	83.9	82.6	78.4	77.1	82.8	82.3	75.4	74.9	86.8	84.2	80.1	77.5	82.0	81.1	79.6	78.8
MPA3D (Ours) 4f	86.4	84.9	81.6	80.1	85.4	84.9	78.7	78.2	87.1	84.1	81.2	78.2	86.7	85.8	84.9	84.0

5.3.3 Comparison with state-of-the-art works

We report the performance of our MPA3D on the WOD validation split against off-the-shelf 3D detectors in Table 5.1. We categorize existing approaches into two groups. The first group contains methods that perform 3D detection using only the current frame, while the second group consists of methods that use multiple-frame inputs. These multi-frame methods extend their single-frame counterparts by concatenating sensor data from previous frames and usually achieve superior performance compared to single-frame methods. Using four-frame inputs, our approach demonstrates superior performance over all existing approaches in the multi-frame category. For instance, we outperform the previous best multi-frame method SAFDNet [185] by 2.2% in overall L1 APH and 2.7% in overall L2 APH.

Additionally, we compare our approach with existing temporal-fusion-based methods in Table 5.2. Notably, these approaches utilize up to 99 previous frames of sensor data, which is substantially more than our four-frame input. Compared to the best temporal fusion method MAD [3], our approach improves the overall L2 AP and APH by 0.2% and 0.4% on validation set and 0.9% and 1.2% on testing set.

Table 5.2. Comparisons with temporal fusion based approaches. The metrics are 3D AP and APH for both L1 and L2 difficulties. We use **bold numbers** to highlight the best results.

	Method	AP L1	APH L1	AP L2	APH L2
<i>Validation</i>	LEF [41]	79.6	79.2	71.4	70.9
	MoDAR [78]	-	-	-	72.5
	MPPNet [22]	81.6	81.1	76.0	74.8
	MSF [39]	82.2	80.7	76.8	75.5
	PTT [54]	82.7	80.7	77.7	75.7
	MAD [3]	85.8	84.2	81.0	79.4
	MPA3D (Ours)	86.4	84.9	81.6	80.1
<i>Testing</i>	3D-MAN [171]	49.6	48.1	44.8	43.4
	MPPNet [22]	81.8	80.6	76.9	75.7
	MSF [39]	83.1	81.7	78.3	77.0
	MAD [3]	86.0	84.3	81.8	80.2
	MPA3D (Ours)	87.2	85.9	83.0	81.6

Table 5.3. Results on WOD test set leaderboard. The metrics are 3D AP and APH for both L1 and L2 difficulties. We use **bold numbers** to highlight the best results.

Method	AP L1	APH L1	AP L2	APH L2
CenterFormer [199]	82.3	80.9	77.6	76.3
BEVFusion [91]	82.7	81.4	77.7	76.3
MSF [39]	83.1	81.7	78.3	77.0
FSD++ [31]	83.5	82.1	78.4	77.1
LoGoNet [75]	83.1	81.8	78.4	77.1
Octopus_Noah	83.1	81.7	78.7	77.3
SEED-L [90]	83.5	82.2	78.7	77.3
LION [89]	83.7	82.4	78.7	77.4
MAD [3]	86.0	84.3	81.8	80.2
MPA3D (Ours)	87.2	85.9	83.0	81.6

Furthermore, Table 5.3 reports our state-of-the-art performance on the WOD test set leaderboard among all online methods that do not use ensembles or test-time augmentation. These superior results validate that our mapping-augmented approach provides richer and more reliable context than traditional temporal fusion, achieving better performance while using significantly fewer frames. This confirms the effectiveness of leveraging high-quality scene reconstruction priors for 3D detection.

Table 5.4. Ablation study on different Mapping Priors. “SWFormer[†]” is the reproduction of SWFormer with our implementation. The baseline models do not use mapping priors (**x**). For each model, the best and second best results are highlighted by **bold** and underlined. Results are reported on a subset of WOD validation split.

Base Model	Surfel	3DGS	Vehicle L2		Pedestrian L2		Cyclist L2		Overall L2	
			AP	APH	AP	APH	AP	APH	AP	APH
SWFormer [†]	x	x	79.2	78.4	73.8	<u>69.3</u>	68.1	67.3	73.7	71.6
SWFormer [†]	✓	x	79.6	78.8	73.5	<u>69.0</u>	<u>69.7</u>	<u>68.8</u>	<u>74.3</u>	<u>72.2</u>
SWFormer [†]	x	✓	80.1	<u>79.1</u>	<u>74.3</u>	69.0	68.3	67.4	74.2	71.9
SWFormer [†]	✓	✓	<u>79.9</u>	79.1	74.3	69.6	70.4	69.4	74.9	72.7
Ours-baseline	x	x	83.7	83.1	79.8	76.3	81.9	81.0	81.8	80.1
Ours-baseline	✓	x	85.3	84.7	<u>80.1</u>	<u>76.7</u>	<u>82.8</u>	<u>81.9</u>	<u>82.7</u>	<u>81.1</u>
Ours-baseline	x	✓	<u>85.5</u>	<u>85.0</u>	80.0	76.4	82.4	81.5	82.6	81.0
Ours-baseline	✓	✓	86.5	85.9	80.5	77.0	82.9	82.0	83.3	81.7

5.3.4 Ablation study

Effectiveness of Mapping Priors. To demonstrate the effectiveness of mapping priors, we conduct several ablation studies and report their results in Table 5.4. We consider two baselines: 1) our implementation of SWFormer [139], denoted as SWFormer[†]; and 2) “Ours-baseline”, which refers to MPA3D without mapping priors while keeping camera and LiDAR inputs. For efficiency, we evaluate these models on a validation subset and report the AP/APH at the L2 difficulty level. We conduct experiments under three different mapping prior settings: with surfel only, with 3DGS only, and with both surfel and 3DGS. For the weaker baseline (SWFormer[†]), adding either surfel or 3DGS provides similar performance gains (*e.g.*, the overall L2 APH improves from 71.6% to 72.2% and 71.9% respectively), though we observe a slight degradation in pedestrian detection. In contrast, for the stronger baseline (Ours-baseline), we observe consistent performance improvements over the baseline model across all categories. Moreover, for both baselines, combining surfel and 3DGS consistently leads to the best performance, as our gated fusion module effectively leverages their complementary strengths.

Conceptually, a mapping prior mostly maps out the static background (since foreground objects are removed), while detection is about finding dynamic foreground objects. The reason

Table 5.5. Ablation study on different fusion strategies. We use **bold numbers** to highlight the best results. Results are reported on a subset of WOD validation split.

Fusion Strategy	Vehicle L2		Pedestrian L2		Cyclist L2		Overall L2	
	AP	APH	AP	APH	AP	APH	AP	APH
Sum	80.3	81.1	71.3	67.7	73.0	72.2	75.2	73.4
Average	82.8	82.0	75.6	71.9	77.9	77.0	78.7	77.0
Concat	83.7	83.1	78.1	74.5	79.2	78.4	80.4	78.7
Gated	86.5	85.9	80.5	77.0	82.9	82.0	83.3	81.7

one helps the other is intuitive: **the map prior allows the model to better separate foreground from background.** This process of elimination is particularly helpful for distant or occluded objects. Fig. 5.1 provides a clear example of this: (Left) Based on sparse LiDAR data alone, the vehicle (yellow box) is ambiguous and difficult to distinguish from the surrounding background noise. (Right) By leveraging the dense map prior, the model can instantly identify all points that align with the known static background (e.g., the road surface, curbs, and vegetation). The remaining, non-matching sparse points—even if they only form a partial outline—are now clearly highlighted as a dynamic foreground object, making the vehicle’s detection (green box) significantly more reliable.

Effectiveness of Gated Fusion. To validate the effectiveness of the proposed gated fusion strategy, we compare it with several common baseline methods: Sum, Average, and Concatenation (Concat). “Sum” involves element-wise addition of per-voxel features from each modality; “Average” computes the mean value across modalities; and “Concat” concatenate the features along the channel dimension without element-wise gating. More details on these variants are provided in the supplementary materials.

As shown in Table 5.5, our gated approach consistently outperforms all other strategies across every class and metric. Notably, “Sum” and “Average” achieve relatively poor performance, likely because they treat all modalities equally; this causes noise or empty features from one modality to indiscriminately corrupt the valid signals from another. In contrast, our method achieves an Overall L2 AP of 83.3%, representing a significant improvement of 2.9% over the

next best method, “Concat” (80.4%). This demonstrates that our adaptive gating mechanism effectively integrates diverse modalities, leading to a substantial boost in detection performance compared to simpler, non-adaptive techniques.

5.4 Conclusion

In this work, we addressed the challenge of improving 3D object detection by replacing resource-intensive HD maps with scalable mapping priors. We introduced Mapping Priors Augmented 3D detection (MPA3D), a novel framework that utilizes automatically generated scene reconstructions, specifically surfels and 3DGS to supplement sparse sensor data. Our core contributions include a scalable data pipeline for automatic prior generation and a Gated Fusion Module to adaptively integrate these priors with LiDAR and camera data, using rich static context to enhance foreground object detection. A mixed-modality training strategy facilitates robustness when priors are unavailable. Extensive experiments on WOD demonstrate that our method achieves new state-of-the-art results, surpassing both multi-frame and complex temporal-fusion methods while using significantly fewer frames. These results validate that leveraging scalable, reconstructed scene priors is a powerful and effective strategy for enhancing 3D perception.

Acknowledgments

Chapter 5, in part, is based on the paper from Yang Fu, Yuliang Zou, Hao Xiang, Xin Huang, Yijing Bai, Chen Song, Weijing Shi, Govind Thattai, Dragomir Anguelov, Mingxing Tan, and Yingwei Li, “Scene Reconstruction as Mapping Priors for 3D Detection” has been accepted and will be published in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2026. The dissertation author is the primary investigator and author of this paper.

Chapter 6

Conclusion and Future Work

This thesis has been motivated by a fundamental disconnect in modern computer vision: despite the remarkable success of 2D foundation models in recognizing and reasoning about visual content, these models remain blind to the geometry, structure, and spatial relationships that define the physical world. At the root of this disconnect lies a data problem — 3D data is expensive, difficult to capture at scale, and insufficient to support the kind of large-scale training that has driven 2D progress. This thesis has addressed this challenge through three interconnected research thrusts: scalable 3D data curation, spatial reasoning with 3D representations, and real-world 3D perception.

Chapter 2: Scalable 3D Data from Reconstruction. The first contribution of this thesis directly attacks the 3D data bottleneck from a reconstruction perspective. Traditional 3D reconstruction pipelines depend heavily on Structure-from-Motion preprocessing — a costly and brittle requirement that limits scalability. This chapter introduced COLMAP-Free 3D Gaussian Splatting, a framework that eliminates this dependency by jointly optimizing camera poses and 3D Gaussian representations from unposed image collections. By removing the preprocessing barrier, this work enables scalable 3D reconstruction from casually captured video, greatly lowering the cost of producing 3D training data. The approach demonstrated competitive reconstruction quality while substantially broadening the range of scenarios from which 3D data can be reliably obtained.

Chapter 3: Scalable 3D Data from Generative Models. The second contribution approaches the data scarcity problem from a complementary direction: synthesis. Rather than capturing or reconstructing 3D data from the real world, this chapter presented HOIDiffusion, a framework that leverages diffusion models to generate realistic 3D hand-object interaction data. Hand-object interaction is a domain where real data is particularly scarce and difficult to annotate, yet critically important for embodied AI. HOIDiffusion generates photorealistic, geometrically consistent interaction data at scale, filling a data gap that reconstruction alone cannot close. Together, Chapters 2 and 3 establish a two-pronged foundation for scalable 3D data acquisition, one rooted in reconstruction and the other in generation.

Chapter 4: Spatial Reasoning — Bridging 2D VLMs to 3D. With a richer base of 3D data established, the thesis then turned to the question of spatial reasoning: how can 2D vision-language models, which have been trained almost exclusively on image-text pairs, be equipped with genuine 3D understanding? This chapter presented two complementary approaches. First, SpatialRGPT was introduced as a framework that augments VLMs with region-level 3D spatial reasoning, enabling models to answer grounded spatial queries such as estimating distances and understanding relative depth. Second, a feature distillation framework was proposed that transfers 3D structural knowledge from 3D-aware representations into 2D VLMs, allowing spatial reasoning capabilities to be injected without requiring full 3D supervision at inference time. These contributions demonstrated that the gap between 2D foundation models and the 3D world can be meaningfully closed through the right representational bridges.

Chapter 5: 3D Perception in the Wild — 6D Object Pose Estimation. The fourth thrust pushed 3D understanding toward real-world deployment, beginning with the problem of category-level 6D object pose estimation. Generalizing pose estimation beyond controlled lab settings — to new object instances, lighting conditions, and capture environments — has remained a persistent challenge. This chapter addressed the problem through two lenses: semi-supervised learning that leverages unlabeled in-the-wild data to improve generalization, and self-supervised geometric correspondence that enables category-level reasoning without dense

annotation. These approaches demonstrated that robust, accurate pose estimation is achievable in unconstrained environments, a key step toward deploying 3D perception systems outside the laboratory.

Chapter 6: 3D Perception in the Wild — Autonomous Driving. The final content chapter scaled 3D perception to one of its most demanding applications: autonomous driving. This chapter explored how scene reconstruction can serve as a structural mapping prior to improve 3D object detection, and investigated the scaling behavior of 3D perception models under large-scale pretraining. The results established that richer geometric priors, combined with data scale, yield meaningful gains in detection robustness and generalization — reinforcing the thesis-wide theme that 3D data and representations are indispensable for real-world visual intelligence.

Future Work

Toward 4D and Dynamic Scene Understanding. Current reconstruction methods, including COLMAP-Free 3DGS, are designed primarily for static scenes. Real-world environments are dynamic, and extending scalable reconstruction to handle moving objects, deforming surfaces, and temporal change remains an important open problem. Jointly modeling scene geometry and scene dynamics — through 4D representations or video-conditioned neural fields — would significantly expand the scope of 3D data that can be curated automatically, with downstream benefits for all tasks explored in this thesis.

Scaling 3D-Aware Foundation Models. The spatial reasoning frameworks developed in Chapter 4 demonstrate that 2D vision-language models can be meaningfully grounded in 3D through targeted representational bridges. However, the scale of 3D supervision available today remains orders of magnitude smaller than what 2D models are trained on. A natural and high-impact direction is to leverage the scalable data curation methods from Chapters 2 and 3 to build truly large-scale 3D pretraining pipelines — and to investigate how model capacity, data diversity, and 3D representation choice interact at scale. The hypothesis, strongly motivated by this thesis,

is that foundation model-scale pretraining on diverse 3D data could unlock a qualitative leap in spatial reasoning and perception robustness, analogous to what scaling achieved in the 2D domain.

Robotics, Embodied AI, and Vision-Language-Action Models. Perhaps the most compelling downstream application of this thesis is robotics, where 3D perception and spatial reasoning are not conveniences but prerequisites. HOIDiffusion is directly applicable to training manipulation policies at scale, and the pose estimation methods from Chapter 5 provide the category-level 6D grounding that generalizable robot grasping demands. Looking further ahead, Vision-Language-Action (VLA) models such as RT-2, OpenVLA, $\pi_{0.5}$, and Gemini Robotics [206, 65, 11, 146] offer a promising path toward generalist robot policies — yet current VLAs are trained predominantly on 2D observations and lack explicit geometric grounding. The region-level 3D spatial reasoning from Chapter 4 could serve as a natural backbone for spatially aware action prediction, and the data pipelines from Chapters 2 and 3 could be repurposed to generate robot demonstration data at scale, enabling the training of VLAs that reason and act with genuine 3D awareness.

The central challenge motivating this thesis — the disconnect between the impressive capabilities of 2D foundation models and the richly three-dimensional world that intelligent systems must ultimately interact with — remains an open problem in computer vision. The contributions presented here, spanning scalable 3D data curation, spatial reasoning, and real-world perception, represent steps toward closing that gap. Much work remains, and the field is moving rapidly. As 3D data becomes more accessible, as representations grow richer, and as reasoning and perception systems mature, the boundary between 2D and 3D intelligence will continue to dissolve. This thesis is offered as a contribution to that ongoing effort.

Bibliography

- [1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.
- [2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Floren-
cia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat,
et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [3] Ben Agro, Sergio Casas, Patrick Wang, Thomas Gilles, and Raquel Urtasun. Mad:
Memory-augmented detection of 3d objects. In *2025 IEEE/CVF Conference on Computer
Vision and Pattern Recognition (CVPR)*, pages 1449–1460, 2025.
- [4] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grund-
mann. Objectron: A large scale dataset of object-centric videos in the wild with pose
annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
Recognition*, pages 7822–7831, 2021.
- [5] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d
question answering for spatial scene understanding. In *proceedings of the IEEE/CVF
conference on computer vision and pattern recognition*, pages 19129–19139, 2022.
- [6] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin,
Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with
versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.
- [7] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin,
Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with
versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.
- [8] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion
IV: control paradigms and data structures*. Spie, 1992.
- [9] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-
nerf: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE/CVF
Conference on Computer Vision and Pattern Recognition*, pages 4160–4169, 2023.

- [10] Frank Biocca. Virtual reality technology: A tutorial. *Journal of communication*, 42(4):23–72, 1992.
- [11] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, et al. $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization. In *9th Annual Conference on Robot Learning (CoRL)*, 2025.
- [12] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015.
- [13] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9044–9053, 2021.
- [14] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European conference on computer vision*, pages 202–221. Springer, 2020.
- [15] Dengsheng Chen, Jun Li, Zheng Wang, and Kai Xu. Learning canonical shape space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11973–11982, 2020.
- [16] Kai Chen and Qi Dou. Sgpa: Structure-guided prior adaptation for category-level 6d object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2773–2782, 2021.
- [17] Pengxin Chen, Zhipeng Luo, Xiaoqi Jiang, Zhangcai Yin, and Jonathan Li. Maps for autonomous driving: Full-process survey and frontiers. *arXiv preprint arXiv:2509.12632*, 2025.
- [18] Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. LI3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26428–26438, 2024.
- [19] Wei Chen, Jinming Duan, Hector Basevi, Hyung Jin Chang, and Ales Leonardis. Point-posenet: Point pose network for robust 6d object pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2824–2833, 2020.
- [20] Wei Chen, Xi Jia, Hyung Jin Chang, Jinming Duan, and Ales Leonardis. G2l-net: Global to local network for real-time 6d pose estimation with embedding vector features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4233–4242, 2020.

- [21] Wei Chen, Xi Jia, Hyung Jin Chang, Jinming Duan, Linlin Shen, and Ales Leonardis. Fs-net: Fast shape-based network for category-level 6d object pose estimation with decoupled rotation mechanism. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1581–1590, 2021.
- [22] Xuesong Chen, Shaoshuai Shi, Benjin Zhu, Ka Chun Cheung, Hang Xu, and Hongsheng Li. Mppnet: Multi-frame feature intertwining with proxy points for 3d temporal object detection. In *European Conference on Computer Vision*, pages 680–697. Springer, 2022.
- [23] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023.
- [24] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3193–3203, 2021.
- [25] Zhenyu Chen, Ronghang Hu, Xinlei Chen, Matthias Nießner, and Angel X Chang. Unit3d: A unified transformer for 3d dense captioning and visual grounding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 18109–18119, 2023.
- [26] 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)*, July 2017.
- [27] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European conference on computer vision (ECCV)*, pages 720–736, 2018.
- [28] Yan Di, Fabian Manhardt, Gu Wang, Xiangyang Ji, Nassir Navab, and Federico Tombari. So-pose: Exploiting self-occlusion for direct 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12396–12405, 2021.
- [29] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- [30] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing Single Stride 3D Object Detector with Sparse Transformer. In *CVPR*, 2022.
- [31] Lue Fan, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Fully Sparse 3D Object Detection. In *NeurIPS*, 2022.

- [32] Lue Fan, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Fsd v2: Improving fully sparse 3d object detection with virtual voxels. *arXiv preprint arXiv:2308.03755*, 2023.
- [33] Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual understanding and reasoning. *arXiv preprint arXiv:2403.11401*, 2024.
- [34] Ross Girshick. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.
- [35] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
- [36] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.
- [37] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. *arXiv*, 2023.
- [38] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3196–3206, 2020.
- [39] Chenhang He, Ruihuang Li, Yabin Zhang, Shuai Li, and Lei Zhang. Msf: Motion-guided sequential fusion for efficient 3d object detection from point cloud sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5196–5205, 2023.
- [40] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- [41] Tong He, Pei Sun, Zhaoqi Leng, Chenxi Liu, Dragomir Anguelov, and Mingxing Tan. Lef: Late-to-early temporal fusion for lidar 3d object detection. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1637–1644. IEEE, 2023.
- [42] Yisheng He, Haibin Huang, Haoqiang Fan, Qifeng Chen, and Jian Sun. Ffb6d: A full flow bidirectional fusion network for 6d pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

- [43] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11632–11641, 2020.
- [44] D Hendrycks. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
- [45] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.
- [46] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.
- [47] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *2011 international conference on computer vision*, pages 858–865. IEEE, 2011.
- [48] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [49] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- [50] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *arXiv preprint arXiv:2307.12981*, 2023.
- [51] Ji Hou, Xiaoliang Dai, Zijian He, Angela Dai, and Matthias Nießner. Mask3d: Pre-training 2d vision transformers by learning masked 3d priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13510–13519, 2023.
- [52] Haifeng Huang, Zehan Wang, Rongjie Huang, Luping Liu, Xize Cheng, Yang Zhao, Tao Jin, and Zhou Zhao. Chat-3d v2: Bridging 3d scene and large language models with object identifiers. *arXiv preprint arXiv:2312.08168*, 2023.
- [53] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023.
- [54] Kuan-Chih Huang, Weijie Lyu, Ming-Hsuan Yang, and Yi-Hsuan Tsai. Ptt: Point-trajectory transformer for efficient temporal 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14938–14947, 2024.

- [55] Tianyu Huang, Bowen Dong, Yunhan Yang, Xiaoshui Huang, Rynson WH Lau, Wanli Ouyang, and Wangmeng Zuo. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22157–22167, 2023.
- [56] Zhening Huang, Xiaoyang Wu, Xi Chen, Hengshuang Zhao, Lei Zhu, and Joan Lasenby. Openins3d: Snap and lookup for 3d open-vocabulary instance segmentation. In *European Conference on Computer Vision*, pages 169–185. Springer, 2025.
- [57] Shun Iwase, Xingyu Liu, Rawal Khrodar, Rio Yokota, and Kris M Kitani. Repose: Fast 6d object pose refinement via deep texture rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3303–3312, 2021.
- [58] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023.
- [59] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *ICCV*, 2021.
- [60] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023.
- [61] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.
- [62] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics (TOG)*, 43(4):1–15, 2024.
- [63] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics (TOG)*, 43(4):1–15, 2024.
- [64] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. *Advances in Neural Information Processing Systems*, 37:80965–80986, 2024.
- [65] 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. *arXiv preprint arXiv:2406.09246*, 2024.

- [66] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [67] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.
- [68] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 2017.
- [69] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *European Conference on Computer Vision*, pages 574–591. Springer, 2020.
- [70] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019.
- [71] Zhaoqi Leng, Guowang Li, Chenxi Liu, Ekin Dogus Cubuk, Pei Sun, Tong He, Dragomir Anguelov, and Mingxing Tan. Lidaraugment: Searching for scalable 3d lidar data augmentations. *arXiv preprint arXiv:2210.13488*, 2022.
- [72] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.
- [73] Chi Li, Jin Bai, and Gregory D Hager. A unified framework for multi-view multi-class object pose estimation. In *Proceedings of the european conference on computer vision (eccv)*, pages 254–269, 2018.
- [74] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024.
- [75] Xin Li, Tao Ma, Yuenan Hou, Botian Shi, Yuchen Yang, Youquan Liu, Xingjiao Wu, Qin Chen, Yikang Li, Yu Qiao, et al. Logonet: Towards accurate 3d object detection with local-to-global cross-modal fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17524–17534, 2023.
- [76] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018.
- [77] Yicheng Li, Qi Li, Tian Guo, Li Wang, Yu Wang, Qinhong Zhang, Yi Ding, Yingfeng Zhang, and Liangjun Zheng. Hdmapnet: An online hd map construction and evaluation

- framework. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4628–4634, 2022.
- [78] Yingwei Li, Charles R Qi, Yin Zhou, Chenxi Liu, and Dragomir Anguelov. Modar: Using motion forecasting for 3d object detection in point cloud sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9329–9339, 2023.
 - [79] Yihan Liao, Yicheng Li, Yinghong Chen, Qinhong Zhang, and Li Zhang. Maptr: Structured modeling and learning for online vectorized hd map construction. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv preprint arXiv:2303.12574.
 - [80] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021.
 - [81] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26689–26699, 2024.
 - [82] Jiehong Lin, Zewei Wei, Zhihao Li, Songcen Xu, Kui Jia, and Yuanqing Li. Dualposenet: Category-level 6d object pose and size estimation using dual pose network with refined learning of pose consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3560–3569, 2021.
 - [83] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
 - [84] Zhi-Hao Lin, Sheng-Yu Huang, and Yu-Chiang Frank Wang. Convolution in the cloud: Learning deformable kernels in 3d graph convolution networks for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1800–1809, 2020.
 - [85] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.
 - [86] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.
 - [87] Jing Liu, Zheng Ding, Tianqi Zhang, Jiaqi Chen, and Jifeng Zhang. Vectormapnet: End-to-end vectorized hd map learning. In *International Conference on Machine Learning (ICML)*, 2023. arXiv preprint arXiv:2303.08785.
 - [88] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7708–7717, 2019.

- [89] Zhe Liu, Jinghua Hou, Xinyu Wang, Xiaoqing Ye, Jingdong Wang, Hengshuang Zhao, and Xiang Bai. Lion: Linear group rnn for 3d object detection in point clouds. *Advances in Neural Information Processing Systems*, 37:13601–13626, 2024.
- [90] Zhe Liu, Jinghua Hou, Xiaoqing Ye, Tong Wang, Jingdong Wang, and Xiang Bai. Seed: A simple and effective 3d detr in point clouds. In *European Conference on Computer Vision*, pages 110–126. Springer, 2024.
- [91] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird’s-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022.
- [92] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023.
- [93] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022.
- [94] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. *arXiv preprint arXiv:2202.07123*, 2022.
- [95] Fabian Manhardt, Gu Wang, Benjamin Busam, Manuel Nickel, Sven Meier, Luca Minicullo, Xiangyang Ji, and Nassir Navab. Cps++: Improving class-level 6d pose and shape estimation from monocular images with self-supervised learning. *arXiv preprint arXiv:2003.05848*, 2020.
- [96] Eitan Marder-Eppstein. Project tango. In *ACM SIGGRAPH 2016 Real-Time Live!*, pages 25–25. 2016.
- [97] 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. *Communications of the ACM*, 2021.
- [98] 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. *Communications of the ACM*, 65(1):99–106, 2021.
- [99] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2901–2910, 2019.
- [100] Jiteng Mu, Weichao Qiu, Adam Kortylewski, Alan Yuille, Nuno Vasconcelos, and Xiao-long Wang. A-sdf: Learning disentangled signed distance functions for articulated shape representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13001–13011, 2021.

- [101] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021.
- [102] Tuan Duc Ngo, Binh-Son Hua, and Khoi Nguyen. Isbnet: a 3d point cloud instance segmentation network with instance-aware sampling and box-aware dynamic convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13550–13559, 2023.
- [103] Phuc Nguyen, Tuan Duc Ngo, Evangelos Kalogerakis, Chuang Gan, Anh Tran, Cuong Pham, and Khoi Nguyen. Open3dis: Open-vocabulary 3d instance segmentation with 2d mask guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4018–4028, 2024.
- [104] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018.
- [105] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
- [106] Teddy Ort. *Autonomous navigation without HD prior maps*. PhD thesis, Massachusetts Institute of Technology, 2022.
- [107] Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Colored point cloud registration revisited. In *ICCV*, 2017.
- [108] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
- [109] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4561–4570, 2019.
- [110] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 815–824, 2023.
- [111] Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 335–342, 2000.
- [112] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European conference on computer vision*, pages 194–210. Springer, 2020.

- [113] Sergey Prokudin, Christoph Lassner, and Javier Romero. Efficient learning on point clouds with basis point sets. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4332–4341, 2019.
- [114] 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*, pages 652–660, 2017.
- [115] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024.
- [116] Mahdi Rad and Vincent Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3828–3836, 2017.
- [117] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [118] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [119] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.
- [120] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021.
- [121] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022.
- [122] Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative vision foundation model reduce all domains into one. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12490–12500, June 2024.
- [123] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10901–10911, 2021.

- [124] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [125] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration. In *IEEE International Conference on Computer Vision Workshops*, 2021.
- [126] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *European Conference on Computer Vision*, pages 125–141. Springer, 2022.
- [127] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.
- [128] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.
- [129] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.
- [130] Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020.
- [131] Guangsheng Shi, Ruifeng Li, and Chao Ma. Pillarnet: Real-time and high-performance pillar-based 3d object detection. In *ECCV*, 2022.
- [132] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5333–5343, 2024.
- [133] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. In *IJCV*, 2023.
- [134] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. In *TPAMI*, 2019.
- [135] Andrea Simonelli, Samuel Rota Buló, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1991–1999, 2019.

- [136] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations, 2020.
- [137] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [138] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.
- [139] Pei Sun, Mingxing Tan, Weiyue Wang, Chenxi Liu, Fei Xia, Zhaoqi Leng, and Dragomir Anguelov. Swformer: Sparse window transformer for 3d object detection in point clouds. In *ECCV*, 2022.
- [140] Pei Sun, Weiyue Wang, Yuning Chai, Gamaleldin Elsayed, Alex Bewley, Xiao Zhang, Cristian Sminchisescu, and Dragomir Anguelov. Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5725–5734, 2021.
- [141] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 699–715, 2018.
- [142] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: A dataset of whole-body human grasping of objects. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 581–600. Springer, 2020.
- [143] Ayça Takmaz, Elisabetta Fedele, Robert W Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. Openmask3d: Open-vocabulary 3d instance segmentation. *arXiv preprint arXiv:2306.13631*, 2023.
- [144] Matthew Tancik, Vincent Casser, Xincheng Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8248–8258, 2022.
- [145] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
- [146] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

- [147] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 292–301, 2018.
- [148] Meng Tian, Marcelo H Ang, and Gim Hee Lee. Shape prior deformation for categorical 6d object pose and size estimation. In *European Conference on Computer Vision*, pages 530–546. Springer, 2020.
- [149] Meng Tian, Marcelo H Ang Jr, and Gim Hee Lee. Shape prior deformation for categorical 6d object pose and size estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, August 2020.
- [150] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018.
- [151] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991.
- [152] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densfusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3343–3352, 2019.
- [153] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16611–16621, 2021.
- [154] Haiyang Wang, Chen Shi, Shaoshuai Shi, Meng Lei, Sen Wang, Di He, Bernt Schiele, and Liwei Wang. Dsvt: Dynamic sparse voxel transformer with rotated sets. In *CVPR*, 2023.
- [155] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019.
- [156] 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.
- [157] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF—: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

- [158] Bowen Wen, Chaitanya Mitash, Sruthi Soorian, Andrew Kimmel, Avishai Sintov, and Kostas E Bekris. Robust, occlusion-aware pose estimation for objects grasped by adaptive hands. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6210–6217. IEEE, 2020.
- [159] Benny Wijaya, Kun Jiang, Mengmeng Yang, Tuopu Wen, Yunlong Wang, Xuewei Tang, Zheng Fu, Taohua Zhou, and Diange Yang. High definition map mapping and update: A general overview and future directions. *arXiv preprint arXiv:2409.09726*, 2024.
- [160] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*, 2023.
- [161] Qi Wu, Janick Martinez Esturo, Ashkan Mirzaei, Nicolas Moenne-Loccoz, and Zan Gojcic. 3dgut: Enabling distorted cameras and secondary rays in gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26036–26046, 2025.
- [162] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4840–4851, 2024.
- [163] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.
- [164] Jinsheng Xiao, Shurui Wang, Jian Zhou, Ziyue Tian, Hongping Zhang, and Yuan-Fang Wang. Mim: High-definition maps incorporated multi-view 3d object detection. *IEEE Transactions on Intelligent Transportation Systems*, 26(3):3989–4001, 2025.
- [165] Xuan Xiong, Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Zhao Hang. Neural map prior for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, 2023.
- [166] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. In *Sensors*, 2018.
- [167] Bin Yang, Ming Liang, and Raquel Urtasun. Hdnet: Exploiting hd maps for 3d object detection. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 146–155. PMLR, 29–31 Oct 2018.
- [168] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2020.

- [169] Jianing Yang, Xuweiyi Chen, Shengyi Qian, Nikhil Madaan, Madhavan Iyengar, David F Fouhey, and Joyce Chai. Llm-grounder: Open-vocabulary 3d visual grounding with large language model as an agent. *arXiv preprint arXiv:2309.12311*, 2023.
- [170] Lixin Yang, Kailin Li, Xinyu Zhan, Fei Wu, Anran Xu, Liu Liu, and Cewu Lu. Oakink: A large-scale knowledge repository for understanding hand-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20953–20962, 2022.
- [171] Zetong Yang, Yin Zhou, Zhifeng Chen, and Jiquan Ngiam. 3d-man: 3d multi-frame attention network for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1863–1872, 2021.
- [172] Zhenpei Yang, Yuning Chai, Dragomir Anguelov, Yin Zhou, Pei Sun, Dumitru Erhan, Sean Rafferty, and Henrik Kretzschmar. Surfelgan: Synthesizing realistic sensor data for autonomous driving. *arXiv*, 2020.
- [173] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *European Conference on Computer Vision*, pages 162–179. Springer, 2024.
- [174] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for Gaussian splatting. *arXiv preprint arXiv:2409.06765*, 2024.
- [175] Yufei Ye, Xueting Li, Abhinav Gupta, Shalini De Mello, Stan Birchfield, Jiaming Song, Shubham Tulsiani, and Sifei Liu. Affordance diffusion: Synthesizing hand-object interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22479–22489, 2023.
- [176] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *CVPR*, 2021.
- [177] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.
- [178] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 516–520, 2016.
- [179] Yuanwen Yue, Anurag Das, Francis Engelmann, Siyu Tang, and Jan Eric Lenssen. Improving 2D Feature Representations by 3D-Aware Fine-Tuning. In *European Conference on Computer Vision (ECCV)*, 2024.
- [180] Yuanwen Yue, Anurag Das, Francis Engelmann, Siyu Tang, and Jan Eric Lenssen. Improving 2d feature representations by 3d-aware fine-tuning. In *European Conference on Computer Vision*, pages 57–74. Springer, 2025.

- [181] Andy Zeng, Kuan-Ting Yu, Shuran Song, Daniel Suo, Ed Walker, Alberto Rodriguez, and Jianxiong Xiao. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 1386–1383. IEEE, 2017.
- [182] Yihan Zeng, Chenhan Jiang, Jiageng Mao, Jianhua Han, Chaoqiang Ye, Qingqiu Huang, Dit-Yan Yeung, Zhen Yang, Xiaodan Liang, and Hang Xu. Clip2: Contrastive language-image-point pretraining from real-world point cloud data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15244–15253, 2023.
- [183] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023.
- [184] Gang Zhang, Junnan Chen, Guohuan Gao, Jianmin Li, and Xiaolin Hu. HEDNet: A hierarchical encoder-decoder network for 3d object detection in point clouds. In *NeurIPS*, 2023.
- [185] Gang Zhang, Junnan Chen, Guohuan Gao, Jianmin Li, Si Liu, and Xiaolin Hu. Safdnet: A simple and effective network for fully sparse 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14477–14486, 2024.
- [186] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.
- [187] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8552–8562, 2022.
- [188] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.
- [189] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, April 2024.
- [190] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021.
- [191] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

- [192] Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d llm: Learning position-aware video representation for 3d scene understanding. *arXiv preprint arXiv:2412.00493*, 2024.
- [193] Dingfu Zhou, Jin Fang, Xibin Song, Chenye Guan, Junbo Yin, Yuchao Dai, and Ruigang Yang. Iou loss for 2d/3d object detection. In *2019 international conference on 3D vision (3DV)*, pages 85–94. IEEE, 2019.
- [194] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. *arXiv preprint arXiv:2310.06773*, 2023.
- [195] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suyu You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024.
- [196] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.
- [197] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks, 2018.
- [198] Yin Zhou, Pei Sun, Yu Zhang, Dragomir Anguelov, Jiyang Gao, Tom Ouyang, James Guo, Jiquan Ngiam, and Vijay Vasudevan. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In *Conference on Robot Learning*, pages 923–932. PMLR, 2020.
- [199] Zixiang Zhou, Xiangchen Zhao, Yu Wang, Panqu Wang, and Hassan Foroosh. Center-former: Center-based transformer for 3d object detection. In *ECCV*, 2022.
- [200] Chenming Zhu, Tai Wang, Wenwei Zhang, Kai Chen, and Xihui Liu. Empowering 3d visual grounding with reasoning capabilities. *arXiv preprint arXiv:2407.01525*, 2024.
- [201] Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering llms with 3d-awareness. *arXiv preprint arXiv:2409.18125*, 2024.
- [202] Liyuan Zhu, Yue Li, Erik Sandström, Shengyu Huang, Konrad Schindler, and Iro Armeni. Loopsplat: Loop closure by registering 3d gaussian splats. *arXiv preprint arXiv:2408.10154*, 2024.
- [203] Menglong Zhu, Konstantinos G Derpanis, Yinfei Yang, Samarth Brahmbhatt, Mabel Zhang, Cody Phillips, Matthieu Lecce, and Kostas Daniilidis. Single image 3d object detection and pose estimation for grasping. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3936–3943. IEEE, 2014.
- [204] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2639–2650, 2023.

- [205] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. *arXiv preprint arXiv:2308.04352*, 2023.
- [206] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.