3D Mapping and Understanding with Learning-based Priors



Wenjing Bian
St. Anne's College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy
Hilary 2025

I desire to know everything about humans.

Abstract

Humans effortlessly use prior knowledge and experience to understand 3D environments from 2D observations. In fields like AR/VR and robotics, we seek to equip vision models with similar capabilities, such as recognising objects, estimating their shapes, and localising the camera. However, the information available at inference time is often ambiguous or incomplete. To bridge this gap, vision models can acquire prior knowledge about objects and environments by learning from data, which enhances their understanding of the 3D world and improves performance on these tasks.

In this thesis, we explore a range of methods for incorporating learning-based priors into 3D vision models, resulting in enhanced accuracy and robustness across multiple related tasks. We begin by focusing on object-level 3D vision. For monocular object reconstruction, we propose a ray-based prior that captures multi-level features, effectively constraining global shape while preserving fine details. Combined with our ray-based 3D representation, the proposed model achieves strong accuracy and generalisation, along with improved inference speed. Next, we introduce a diffusion-based framework that leverages visual priors from various sources to enable category-agnostic 3D object detection. This approach demonstrates strong in-domain performance and exhibits robustness to novel object categories in challenging, in-the-wild scenarios.

Shifting to scene-level 3D vision, we address the joint optimisation of scene structure and camera poses in both known and unknown pose settings. When pose information is unavailable, we incorporate a geometric prior from a monocular depth estimator to regularise both geometry and relative camera poses, resulting in more robust joint optimisation. In scenarios where reliable pose initialisations are available, we further refine both poses and scene geometry by introducing an alternative parameterisation for global trajectory constraints and incorporating correspondence priors through an epipolar geometry loss. Finally, we propose multiple geometric priors to enhance scene coordinate regression, leading to improved performance in reconstruction and visual relocalisation.

Keywords – Multi-view Geometry, 3D Reconstruction, 3D Detection, Visual Relocalisation, Diffusion

Declaration

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

Wenjing Bian, April 2025.

Acknowledgement

First and foremost, I would like to express my deepest gratitude to Prof. Victor Adrian Prisacariu for giving me my first opportunity to step into the field of computer vision and for guiding me throughout this journey. His mentorship has been pivotal in shaping my development, and his unique perspectives have been a constant source of inspiration. He has always been patient, open to my ideas, and supportive, helping me build confidence and develop essential problem-solving skills. I am also very grateful to Prof. Andrea Vedaldi, whose wisdom inspires me and who generously supports my work. Furthermore, I would like to thank my undergraduate tutors, Prof. David Murray and Prof. Budimir Rosic, for nurturing my interest in research and helping me build a strong foundation for further studies.

I would like to sincerely thank all my collaborators, including Zirui Wang, Kejie Li, and Jiawang Bian at AVL, as well as Eric Brachmann, Tommaso Cavallari, and Axel Barroso-Laguna at Niantic, for their invaluable insights and support in my research. Throughout this bittersweet journey, we explored fascinating ideas, made meaningful discoveries, and overcame numerous challenges together.

I am also grateful to my mentor and colleagues at Meta Reality Lab—Omkar M. Parkhi, Kang Zheng, Guruprasad Somasundaram, Shangyi Cheng, Santosh Janardhan, Nikhil Raina, and Carl Ren—for their support both during and after my internship. A special thank you to Michael Hobley for his help in refining my papers, to Shuda Li and Gratianus Wesley Putra Data for their support during my 4^{th} year project, and to Theo Costain for taking care of the lab servers.

Having spent over eight years at St Anne's College, I am deeply grateful to all the staff members who have been like family to me. I will always treasure the memories of meals in the dining hall and on the lawn, peaceful hours spent punting, and days and nights in the library. My heartfelt thanks go to my friends for their companionship and for sharing so many joyful moments along the way.

My parents and grandparents have given me unwavering support and care throughout the years I have dedicated to studying. Their love is my constant source of motivation and strength through life's challenges. Though distance and time separate us, our hearts remain deeply connected. I hope my grandfather would have been proud of me if he had witnessed me complete my degree.

Contents

1 Introduction **6**
- 1.1 Motivation . . . 6
- 1.2 Key Contributions . . . 8
  - 1.2.1 Learning Priors for Object-Level 3D Vision . . . 8
  - 1.2.2 Learning Priors for Scene-Level 3D Vision . . . 11
- 1.3 Thesis Outline . . . 13
- 1.4 List of Publications . . . 13

2 Literature Review **15**
- 2.1 Representations for 3D Geometry . . . 15
  - 2.1.1 Explicit 3D Representations . . . 16
  - 2.1.2 Implicit 3D Representations . . . 17
  - 2.1.3 Hybrid 3D Representations . . . 20
- 2.2 Image-based 3D Reconstruction . . . 21
  - 2.2.1 Single-view Object Reconstruction . . . 21
  - 2.2.2 Multi-view Scene Reconstruction . . . 22
- 2.3 Diffusion in 3D Vision . . . 24
  - 2.3.1 3D Detection with Diffusion . . . 25
  - 2.3.2 Unconditional 3D Generation with Diffusion . . . 25

3 Ray-ONet: Efficient 3D Reconstruction From A Single RGB Image **27**

4 CatFree3D: Category-agnostic 3D Object Detection with Diffusion **43**

5 NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior **61**

6 PoRF: Pose Residual Field for Accurate Neural Surface Reconstruction **80**

7 Scene Coordinate Reconstruction Priors **99**

8 Conclusion **118**
- 8.1 Future Work . . . 119

Chapter 1

Introduction

1.1 Motivation

Interpreting the 3D world from 2D projections comes naturally to humans, whether through direct visual perception or by observing images and videos. This ability largely stems from the rich prior knowledge we carry. For instance, our familiarity with common object shapes helps us infer their size and distance, even when viewed from just a single point of view. With access to multiple views, we can integrate information across views to refine our spatial understanding. For computer vision models to achieve similar capabilities, incorporating high-level visual priors can greatly improve their ability to interpret and navigate complex environments, expanding their potential in fields like augmented reality and robotics.

Historically, 3D vision has relied heavily on explicit assumptions to address the inherent ambiguities of interpreting 2D images. These assumptions were largely grounded in well-established principles of geometry, optics, and the physical structure of the real world. For example, geometric multi-view consistency is a foundational assumption in reconstruction tasks involving static scenes and objects [1, 2]. The assumption of Lambertian surfaces enables the use of brightness constancy

constraints in classical methods such as optical flow and stereo matching [3, 4]. Moreover, the "Manhattan world" assumption—where environments are predominantly composed of orthogonal surfaces—has been widely adopted to simplify scene understanding in urban settings [5, 6].

Over the past decade, deep learning has driven substantial progress across a wide range of vision tasks. The emergence of large foundation models [7, 8, 9, 10] and increasingly powerful feature backbones [11, 12, 13, 14] has enabled the extraction of rich learned priors, facilitating advancements in diverse applications such as object detection [15, 16], image segmentation [17], and content generation across modalities—including images [10, 18], videos [19], 3D scenes [20, 21], and dynamic 4D environments [22, 23].

However, effectively leveraging these learning-based priors in 3D vision tasks remains a complex challenge. Monocular depth estimation, for instance, often suffers from scale ambiguities and geometric distortions, while features extracted by backbone networks often lack robustness when transferred across diverse domains. Additionally, inconsistencies between different priors, or between learned priors and traditional explicit assumptions, can undermine model performance. A key challenge lies in identifying which priors are most beneficial in a given context and, equally importantly, determining when and how to rely on them effectively.

This thesis explores a range of methods for integrating learning-based prior knowledge about 3D objects and scenes into 3D vision models. It shows how such priors can improve model performance across a range of tasks, including 3D object reconstruction (Chapter 3), 3D object detection (Chapter 4), joint scene-pose optimisation (Chapter 5 and Chapter 6), and visual relocalisation (Chapter 7).

The following section outlines the specific tasks addressed in this thesis, their respective problem settings, and our key contributions to each.

1.2 Key Contributions

Following the broad motivation outlined in the previous section, we now detail the specific problems addressed in this work. We begin in Section 1.2.1 by exploring how various priors can be incorporated into object-level 3D vision tasks, with a focus on 3D object reconstruction (Section 1.2.1.1) and 3D object detection (Section 1.2.1.2). We then turn our attention to scene-level 3D reconstruction in Section 1.2.2, where we investigate the role of learning-based priors in enhancing two key downstream tasks: joint pose and scene optimisation (Section 1.2.2.1) and scene coordinate regression (Section 1.2.2.2).

1.2.1 Learning Priors for Object-Level 3D Vision

In object-level vision tasks, limited observations often make accurate inference challenging. In such scenarios, visual priors related to an object's shape, category, and appearance can significantly enhance object understanding. In the following sections, we focus on two specific tasks—monocular object reconstruction and monocular 3D object detection—and demonstrate how incorporating various learning-based priors can substantially improve performance.

1.2.1.1 Object Reconstruction with Ray-based Priors

Inferring 3D object geometry from a single image is inherently ill-posed due to the absence of depth information. However, humans can effortlessly perceive 3D shapes by leveraging prior knowledge of object types and forms, even without multi-view cues.

Early deep learning-based methods allow us to incorporate global prior knowledge for specific object categories through training data [24, 25, 26, 27]. However, the reconstructions often lack details, and these approaches struggle with general-

isation across other object categories that do not occur in the training set. More advanced approaches [28, 29] utilise local features extracted from the input image to provide high-level visual priors for each 3D query point. While effective to some extent, relying exclusively on local features limits the model's ability to form a coherent global understanding of the object, frequently leading to distortions in unobserved regions. Moreover, because local features remain constant for all 3D points along the same camera ray, they introduce ambiguity and hinder precise reasoning.

In terms of 3D representations, explicit formats such as voxels, meshes, and point clouds offer fast inference but are limited by resolution constraints. Occupancy Networks (ONet) [26] and DeepSDF [25] adopt continuous representations, enhancing 3D expressiveness without increasing model size. However, these methods come with significantly higher inference costs, as they require querying each 3D point individually, resulting in a complexity of $O(N^3)$.

In Chapter 3, we introduce an approach for learning 3D object priors that integrates both global structural cues and local photometric details. Rather than learning features for individual 3D query points, we propose a ray-based representation, where local features are learned per ray passing through a specific pixel in the image, alongside a per-image global feature. We demonstrate that our method not only surpasses previous state-of-the-art approaches but also exhibits stronger generalisation to unseen object categories and real-world images, despite being trained exclusively on synthetic data. Additionally, the ray-based representation achieves a complexity of $O(N^2)$, significantly reducing inference costs compared to ONet while maintaining a similar model size.

1.2.1.2 Category-agnostic Object Detection with Geometric Priors

Detecting 3D objects from a single image remains a challenging task, primarily due to the high complexity of predicting 9-DoF 3D bounding boxes and the limited availability of diverse training data. Existing methods [30, 31, 32, 16, 33] typically train models to directly map image inputs to 3D box parameters. However, these approaches tend to generalise poorly beyond the indoor static environments and urban traffic scenes—domains where annotated data is more readily available. A key limitation is their reliance on handcrafted or learned anchors or queries for bounding box regression, which can hinder adaptability across different scenarios and object distributions. Additionally, many of these methods depend on object category labels during training, limiting their ability to handle the wide variety of objects found in unconstrained, real-world settings.

In Chapter 4, we present a category-agnostic 3D object detection model that refines randomly initialised 3D bounding boxes toward the target through a diffusion-based denoising process. In contrast to prior approaches, we decouple 3D detection task from 2D detection and depth estimation, instead conditioning the model on multiple visual priors—including image features, 2D bounding boxes, depth maps, and camera intrinsics. This design simplifies the detection task and enhances the generalisation capabilities of the model. By leveraging the diffusion process for box estimation, our method eliminates the need for region proposals, resulting in a more streamlined pipeline capable of producing multiple candidate predictions for each object. Empirically, our model achieves superior in-domain performance compared to previous state-of-the-art methods and demonstrates robust generalisation across diverse object categories and datasets.

1.2.2 Learning Priors for Scene-Level 3D Vision

Compared to object-level priors, which often make strong assumptions about category-specific shape or structure, priors for complex environments tend to be more general, encompassing properties such as surface smoothness and spatial arrangements. In the following, we explore how learning-based priors can be incorporated into scene-level tasks. Specifically, we first explore how geometric and pose priors can be used to regularise joint scene reconstruction and camera pose estimation. We then investigate the use of geometric priors to enhance scene coordinate regression.

1.2.2.1 Joint Scene and Pose Optimisation with Priors

In recent years, Neural Radiance Field (NeRF) [34] and 3D Gaussian Splatting (3DGS) [35] have become widely adopted for their ability to generate high-quality, photorealistic novel views. However, the original NeRF and 3DGS still depend on traditional Structure from Motion (SfM) [2] to precompute camera poses, which are required for scene optimisation. SfM, especially in large-scale scenes with many frames, can be computationally intensive and prone to failure in textureless regions or environments with repetitive patterns. Errors in pose estimation from SfM inevitably propagate into the NeRF or 3DGS optimisation, degrading final performance. To address this, several works [36, 37, 38] have explored jointly optimising camera poses and the NeRF model. While this removes the dependency on external pose estimators, such methods often struggle in cases with large rotational motion and may overfit to image appearance without learning correct 3D geometry. Even when initialised with approximate poses from external estimators, NeRF typically finds it difficult to further refine camera poses, limiting overall reconstruction accuracy.

In this thesis, we propose to incorporate learning-based priors into the joint

optimisation of camera pose and scene representation across two distinct scenarios. In Chapter 5, we assume that no initial camera poses are provided for NeRF optimisation, and we leverage off-the-shelf depth estimation to impose geometric constraints on NeRF reconstruction and between adjacent frames, which results in improved view synthesis quality and pose accuracy. In Chapter 6, assuming pose initialisation is provided by external estimators such as COLMAP or ARKit, we introduce a Pose Residual Field, which (i) offers greater robustness than the conventional pose parameter optimisation due to parameter sharing, and (ii) leverages global information over the entire sequence. Additionally, we incorporate an epipolar geometry loss as a prior on image correspondences. Together, these components enhance both the reconstruction quality and pose accuracy.

1.2.2.2 Scene Coordinate Regression with Geometric Priors

Scene Coordinate Regression (SCR) [39, 40] is a key approach in visual localisationhas also been successfully applied to structure-from-motion (SfM) [41]. Recent variants [42, 43, 44] use neural networks to learn a mapping from image pixels to 3D world coordinates. Thses SCR models rely on dense, implicit triangulation, which requires strong multiview constraints. However, when such constraints are insufficient due to factors like textureless regions or repetitive patterns, even the most advanced SCR frameworks [44, 45] can produce artefacts such as distortions and outliers, leading to reduced accuracy in SfM and relocalisation tasks.

In Chapter 7, we introduce several geometric priors to regularise the mapping stage of SCR models. These priors include a hand-crafted prior over depth distribution and a learned prior that captures plausible indoor scene layouts using a point cloud diffusion model. By reformulating SCR training as a maximum likelihood learning problem,these priors can be seamlessly integrated into existing SCR frameworks such as ACE [44], GLACE [45], and ACE0 [41]. Experimental

results demonstrate that incorporating these priors improves the quality of the reconstructed 3D geometry, leading to higher registration rates in SfM and better relocalisation accuracy, without significantly increasing training time or affecting inference efficiency.

1.3 Thesis Outline

This thesis follows an integrated format[1], in which the core content of several chapters (Chapter 3 to Chapter 7) is presented in the form originally prepared for publication.

The structure of the thesis is as follows: Chapter 2 reviews related literature, while Chapter 3 to Chapter 7 detail the primary research contributions, each corresponding to an accepted publication or a manuscript currently under review. Finally, Chapter 8 summarises the key findings and outlines potential future research directions.

1.4 List of Publications

Chapter 3: "Ray-ONet: Efficient 3D Reconstruction From A Single RGB Image". **Wenjing Bian**, Zirui Wang, Kejie Li, Victor Prisacariu. In *British Machine Vision Conference (BMVC)*, 2021.

Chapter 4: "CatFree3D: Category-agnostic 3D Object Detection with Diffusion". **Wenjing Bian**, Zirui Wang, Andrea Vedaldi. In *International Conference on 3D Vision (3DV)*, 2025.

Chapter 5: "NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior". **Wenjing Bian**, Zirui Wang, Kejie Li, Jia-Wang Bian, Victor

[1]https://www.mpls.ox.ac.uk/graduate-school/information-for-postgraduate-research-students/submitting-your-thesis/guidance-for-submitting-an-integrated-thesis

Adrian Prisacariu. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Chapter 6: "PoRF: Pose Residual Field for Accurate Neural Surface Reconstruction". Jia-Wang Bian, **Wenjing Bian**, Victor Adrian Prisacariu, Philip Torr. In *International Conference on Learning Representations (ICLR)*, 2024.

Chapter 7: "Scene Coordinate Reconstruction Priors". **Wenjing Bian**, Axel Barroso-Laguna, Tommaso Cavallari, Victor Adrian Prisacariu, Eric Brachmann. In *International Conference on Computer Vision (ICCV)*, 2025.

Publication Not Included

⇒ **"CrossScore: Towards Multi-View Image Evaluation and Scoring".** Zirui Wang, **Wenjing Bian**,Victor Prisacariu. In *European Conference on Computer Vision (ECCV)*, 2024

Abstract: CrossScore is a Cross-Reference image quality assessment method that evaluates an image by comparing it with multiple views of the same scene through cross-attention, eliminating the need for a pre-aligned ground truth image.

Chapter 2

Literature Review

This chapter provides an overview of prior work relevant to the research presented in this thesis. It covers both the foundational methods that form the basis of our approach and the more recent advances upon which we build, highlighting the main distinctions and novelty of our approach relative to existing work.

We review the literature in three key areas: Section 2.1 covers commonly used 3D representations, their advantages, associated tasks, and the ones we adopted for each work in the thesis; Section 2.2 discusses image-based 3D reconstruction for objects and scenes; and finally, Section 2.3 reviews diffusion models for 3D vision taks.

2.1 Representations for 3D Geometry

The choice of 3D representation is a fundamental aspect of computer vision and graphics, shaping how 3D data is captured, analysed, visualised, and interacted with. It plays a vital role across a wide range of domains, including robotics, AR/VR, medical imaging, and beyond.

In the following, we classify 3D representations into three categories based

on their underlying formulation—explicit, implicit, and hybrid. We then provide a focused introduction to several representative types within each category that are particularly relevant to the research scope of this thesis, highlighting their respective advantages for specific tasks.

2.1.1 Explicit 3D Representations

Explicit representations of 3D geometry involve discretely sampling surfaces or volumes within 3D space, with the sampled data explicitly stored. Common forms of explicit representation include point clouds, meshes, voxel grids, and depth maps. In the following paragraphs, we provide a focused introduction to point clouds and depth maps, as these are most relevant to the context of this thesis.

Point Clouds A point cloud is a collection of 3D points in space, where each point corresponds to a specific location on a surface. Beyond spatial coordinates, individual points may also carry additional attributes such as colour, intensity, and confidence. Point cloud data is typically acquired using sensors like LiDAR, RADAR, and depth cameras. It is also commonly used as the output representation in Structure-from-Motion (SfM) [2, 46, 47] and Simultaneous Localisation and Mapping (SLAM) [48, 49, 50, 51] algorithms, where it is used to reconstruct environments for a wide range of downstream applications. Point clouds are widely adopted due to the ease of acquisition using commodity sensors and their lower memory requirements compared to other explicit representations, such as voxels or meshes. Deep-learning models have been developed to process point clouds using pointwise operations and global pooling mechanisms to maintain permutation invariance [52, 53]. Furthermore, more advanced backbone architectures [14, 54, 55] have been proposed to effectively encode point cloud data for tasks such as 3D object detection, semantic segmentation, and scene understanding. In Chapter 7, we

propose a diffusion-based point cloud generation model that serves as a geometric prior for scene coordinates during inference.

Depth Maps Depth maps represent the distance from the camera to the closest surface in the scene at each pixel. Traditionally, depth maps can be obtained through approaches such as stereo cameras [4] and structured light [56]. Eigen *et al.* [57] pioneered the use of convolutional neural networks (CNNs) for monocular depth estimation, marking a significant shift in the field. Subsequent works [58, 59, 60] have significantly improved performance by leveraging diverse datasets, enhanced network architectures, advanced loss functions, and optimised training strategies. However, monocular depth estimation remains challenged by scale ambiguity and temporal inconsistency across frames. In Chapter 5, we incorporate monocular depth maps into the task of 3D reconstruction with unknown camera poses using NeRF. The depth maps serve as geometric constraints, regularising both the scene geometry and inter-frame camera poses. To address the inherent scale ambiguity of monocular depth, we introduce a per-frame distortion parameterisation, which is jointly optimised alongside the NeRF representation and camera poses.

2.1.2 Implicit 3D Representations

An implicit 3D representation encodes the 3D object or scene indirectly within the parameters of a learned model rather than explicitly storing geometry as a discrete collection of points, meshes, or voxels. This approach offers the advantage of maintaining a constant memory footprint, independent of the scale and complexity of the scene. Additionally, it removes the need to predefine a fixed spatial resolution. In this thesis, we focus on neural implicit representations, including Occupancy Field, Signed Distance Functions, Neural Radiance Fields and Scene

Coordinates, which are discussed in detail in the following paragraphs.

Occupancy Field An occupancy field [26] models 3D geometry as a continuous function that assigns to each point in space a probability of being occupied, with values ranging from 0 to 1. This probability can be interpreted as the point's opacity. To extract a surface from the occupancy field, a threshold is applied to define a decision boundary that separates occupied from empty regions. The resulting isosurface can then be extracted using algorithms such as marching cubes [61] or marching tetrahedra [62], which operate over a discretized 3D grid.

Signed Distance Functions A signed distance function (SDF) [25] provides a continuous representation of 3D geometry by assigning each point in space a scalar value equal to its distance to the nearest surface, with the sign indicating whether the point is inside or outside the object. The surface corresponds to the zero level set of the SDF and can be extracted using methods such as sphere tracing [63] or marching cubes [61]. SDFs have also been successfully integrated with density fields and volumetric rendering techniques [64, 65, 66] to recover high-quality surfaces from 2D images.

Neural Radiance Field Neural Radiance Field (NeRF) [67] maps 3D coordinates and 2D viewing directions to the volume density and RGB colour of a point using a Multi-layer Perceptron (MLP). This allows NeRF to capture view-dependent effects and lighting, enabling photorealistic view synthesis. Many extensions of NeRF have been developed to accelerate training and inference through modified representations [68, 69, 70, 71]. Other improvements include efforts to make NeRF more generalisable [72, 73, 74], eliminating the need for per-scene optimisation, optimising NeRF jointly with camera parameters [36, 37, 75], applying NeRF to dynamic scenes [76, 77, 78], and scaling it for large, unbounded

scenes [79, 80, 81]. Additionally, incorporating regularisation into NeRF optimisation [82, 83, 84] helps to reduce artefacts in sparse view reconstructions. Another line of work, such as NeuS [64] and UNISURF [66], proposes variants of NeRF that improve surface reconstruction by modifying the volume rendering process and introducing geometric regularisation. A comprehensive survey of NeRF-related research can be found in [85]. In Chapter 5, we utilise NeRF with unknown-pose images for novel view synthesis and introduce a depth prior to regularise its geometry. This approach not only improves the quality of 3D reconstruction and view synthesis but also helps constrain the relative poses between adjacent frames, leading to improved pose optimisation. In Chapter 6, we build upon NeuS by refining camera poses using a proposed Pose Residual Field, which leverages global information through shared parameters. Additionally, we incorporate an epipolar geometry loss, with the combined approach further improving both pose accuracy and reconstruction quality.

Scene Coordinates Scene Coordinate Regression (SCR) was initially proposed for camera relocalisation in small indoor environments using RGB-D images, with scene-specific training [39]. Since then, it has been extended to support on-the-fly adaptation [86, 87, 88], RGB inputs [89, 90], large-scale environments [91, 92, 45], and scene-agnostic pre-training [44]. More recently, it has also been applied to structure-from-motion (SfM) via self-supervised training [41]. SCR models learn a mapping from image pixels to their corresponding 3D world coordinates, enabling dense 2D-3D correspondences. This mapping can be implemented using random forests [86, 93, 90] or neural networks [42, 94, 89]. Given images and known camera poses from a scene, the model is typically trained using a reprojection loss. Once trained, an SCR model can process a new input image to predict 2D-3D correspondences, from which camera pose can be estimated using a Perspective-

n-Point (PnP) solver [95] and RANSAC [96]. However, SCR relies on strong multiview constraints during training. When training views are sparse or ambiguous, the predicted scene coordinates may suffer from distortions or outliers, degrading relocalisation and SfM performance. In Chapter 7, we propose to incorporate geometric priors into SCR training to regularise predicted scene coordinates. Through experiments with several state-of-the-art SCR frameworks, we demonstrate that these priors improve reconstruction quality and lead to higher SfM registration rate and higher relocalisation accuracy.

2.1.3 Hybrid 3D Representations

Explicit representations can directly represent the 3D shape but are constrained by their resolution and memory requirements, whereas implicit representations can handle large-scale scenes but often require time-consuming post-processing to extract a 3D structure. Hybrid 3D representations [97, 98, 99, 100, 101, 35, 102] combine explicit and implicit methods to leverage the strengths of both while addressing their respective limitations. For example, CoReNet [97] integrates voxel grids with implicit volumes by predicting offsets for each grid, achieving translation equivariance and maintaining a constant memory footprint. 3D Gaussian Splatting (3DGS) [35] represents a scene as a collection of 3D Gaussians, where each Gaussian encodes a continuous density function parameterised by its position, covariance, colour, and opacity, enabling high-fidelity and efficient rendering.

In Chapter 3, we propose a ray-based representation that combines implicit representation on the image plane with explicitly sampled points along the ray direction. Unlike implicit occupancy networks (ONet) [26], which query 3D points in space during inference, our approach queries rays on the image plane, reducing inference time complexity from $O(N^3)$ to $O(N^2)$. Importantly, this efficiency gain is achieved without increasing the model size compared to ONet.

2.2 Image-based 3D Reconstruction

Reconstructing 3D objects and scenes from 2D images has been a long-standing challenge in computer vision, with a wide range of applications in fields like AR/VR, robotics, navigation, and medical imaging. In the following sections, we review the literature on 3D reconstruction within two primary settings: object-level reconstruction from a single view and scene-level reconstruction from multiple images. The former aligns with the problem addressed in Chapter 3, while the latter pertains to the methodologies explored in Chapter 5 and Chapter 6.

2.2.1 Single-view Object Reconstruction

Reconstructing the geometry of an object from a single view is an ill-posed problem, as a single image provides limited information, making it challenging to infer absolute scale and accurately reconstruct occluded or unobserved regions. Traditionally, this challenge has been addressed using "shape-from-X" techniques, where "X" refers to cues like shading [103, 104], silhouette [105, 106], shadows [107, 108], and texture [109, 110], all of which rely on strong assumptions about shape or lighting conditions. Early works like [111, 112] used learning-based approaches to reconstruct simple geometric structures.

Deep learning has enabled rich priors about object shape and appearance to be learned implicitly through neural networks. Approaches such as 3D-R2N2 [24], PSGN [113], and Pixel2Mesh [114] have focused on object-centric 3D reconstruction, employing representations like voxels, point clouds, and meshes. However, these methods often suffer from a lack of detail in the reconstructed shapes. To address this, more recent approaches have incorporated implicit representations [26, 25], local feature priors [28, 29, 98, 115], and adopted view-centric predictions [72], enabling the model to capture finer object details. Recent advances

in diffusion models enable rich geometric information to be extracted from pre-trained image diffusion models such as Stable Diffusion [116], which enables more diverse objects with richer details and textures to be reconstructed from a single image, pushing the boundaries of what can be inferred from limited visual input [20, 117, 118].

In Chapter 3, we propose the integration of multi-level image features into a ray-based prediction model, achieving strong reconstruction performance with detailed object structures, while also demonstrating good generalisation across object categories and towards real-world images.

2.2.2 Multi-view Scene Reconstruction

Traditionally, 3D scene reconstruction from multi-view images has been achieved through Structure-from-Motion (SfM) [119, 1, 2], which typically involves identifying keypoints or features in images, establishing correspondences across images, estimating camera poses, and obtaining a sparse point cloud via triangulation and bundle adjustment [120, 121]. Multi-View Stereo (MVS) methods [122, 123] can then be applied to generate dense depth maps by matching patches across neighbouring images, which are subsequently fused into a single, consistent dense 3D point cloud. However, traditional MVS approaches often struggle in challenging scenarios, such as textureless regions, repetitive patterns, or when only sparse views are available, resulting in incomplete reconstructions or artefacts. In recent years, learning-based MVS approaches [124, 125, 126] have been proposed to address the limitations of traditional methods by leveraging learned priors, improving robustness and reconstruction quality in difficult conditions.

Approaches such as NeRF [67], NeuS [64], and 3DGS [35] offer an alternative to traditional MVS methods for generating dense 3D models and producing high-quality novel view synthesis. While NeRF initially relied on SfM algo-

rithms (*e.g.,* COLMAP) to estimate image poses during pre-processing, subsequent works [37, 36] removed this requirement in certain limited scenarios by jointly optimising poses and NeRF, enabling an end-to-end differentiable pipeline for multi-view reconstruction from raw images. However, NeRF was originally designed as a per-scene optimisation method without any prior knowledge of geometry, leading to shape-radiance ambiguity [79]. The unknown poses introduce an additional layer of ambiguity, which causes the optimisation to easily fall into local minima. Several approaches have been proposed to address this challenge, such as incorporating additional pixel matches [127], using generalisable features [128], or making assumptions about the camera trajectory [129]. In Chapter 5, we introduce a method that incorporates monocular depth priors into unknown-pose NeRF optimisation. This regularises both the NeRF geometry and relative poses, resulting in a more accurate and robust pose estimation and 3D reconstruction. In Chapter 6, we propose a Pose Residual Field for pose refinement integrated into NeuS [64], which enhances both pose accuracy and the quality of 3D reconstruction.

In contrast to NeRF and 3DGS which construct 3D scenes that can be used for novel view synthesis, scene coordinate regression [39, 42, 130] learns the mapping between 2D pixels and 3D world coordinates, primarily used for camera relocalisation. However, by aggregating predicted scene coordinates across multiple viewpoints, SCR can also serve as an alternative method for 3D reconstruction. Since SCR models are typically trained using a reprojection loss for each scene, their performance strongly depends on the availability of multi-view constraints. In scenarios where such constraints are limited, even the most advanced methods [44, 45] often exhibit reconstruction artefacts such as outliers and geometric distortions. In Chapter 7, we propose to regularise scene coordinates with reconstruction priors, including a hand-crafted prior based on depth distributions and a learned prior from a diffusion model that captures general indoor scene geometry.

These priors improve the reconstruction quality of scene coordinates and therefore improve the performance of downstream tasks.

2.3 Diffusion in 3D Vision

Diffusion probabilistic models [131, 132, 133] are a class of latent variable models trained by optimising the variational lower bound. These models approximate the posterior via a forward diffusion process modelled as a Markov chain, where Gaussian noise is gradually added to the latent variables over T steps according to a predefined variance schedule, eventually transforming the data into pure noise. The reverse process is learned by training a model to predict the noise added at each step. During inference, the model reconstructs the original data by iteratively denoising from random noise [131, 134].

Diffusion models have proven highly effective at generating high-quality images through this gradual denoising process [10, 18, 135]. They have also been successfully applied to a range of 2D vision tasks, including video generation [136, 137, 138], image segmentation [139, 140, 141], object detection [142], and depth estimation [143, 144]. However, data for 3D vision tasks is relatively scarce compared to 2D data, leading many researchers to focus on transferring pretrained 2D image diffusion models to 3D generation, often in combination with NeRF [145, 146, 20]. These approaches, though promising, tend to be computationally expensive due to the need for per-scene optimisation, and the 2D priors used are not inherently multi-view consistent. Other works has explored applying diffusion models directly to 3D representations, for tasks such as unconditional 3D generation [147, 148, 149], 3D generation conditioned on images [150, 151, 117] or text [152, 153], 3D editing [154, 155] and 3D object detection [156, 157, 158]. A more comprehensive overview of developments on 3D vision tasks with diffusion can be found in [159].

In the following sections, we review the application of diffusion with two specific tasks, 3D detection and unconditional 3D generation, which are relevant to Chapter 4 and Chapter 7 respectively.

2.3.1 3D Detection with Diffusion

3D object detection aims to identify objects and estimate bounding boxes that enclose them within a 3D space. It is widely applied in areas such as AR/VR, autonomous driving, and robotics. Depending on the type of input data, 3D detection approaches can be broadly classified into three categories: point cloud-based methods [160, 161], image-based methods [30, 32], and fusion-based methods, which combine information from both point clouds and images [162, 163].

In recent years, diffusion-based 3D detection methods have emerged [164, 156, 165, 157]. Unlike earlier approaches that rely on regional proposals or handcrafted anchors with pre-defined dimensions and positions [15, 166], the diffusion process enables the model to recover the parameters of target bounding boxes from random noise, simplifying the process. Moreover, the stochastic nature of diffusion allows the number of predicted boxes at inference time to vary independently from training, offering greater flexibility [142, 165]. In Chapter 4, we adopt this diffusion-based approach and introduce a confidence prediction head to select the best prediction from multiple candidates. We apply this model to image-based diffusion with additional geometric prompts, resulting in a detection model with strong generalisation capabilities.

2.3.2 Unconditional 3D Generation with Diffusion

Unconditional 3D generation aims to synthesise 3D models without relying on external conditioning signals such as images, text prompts, or partial observa-

tions. Instead, models learn to capture and reproduce the underlying distribution of shapes purely from training data, enabling the generation of novel yet plausible 3D structures. This capability has broad applications in fields such as 3D content creation, gaming, virtual reality, and simulation. Recent work has explored unconditional 3D generation across various 3D representations, including point clouds [147, 167], Neural Radiance Fields [149, 168, 169], Signed Distance Fields [148, 170], and 3D Gaussian Splatting [171].

Moreover, since diffusion models are closely related to score matching [172, 173], where the estimated noise is proportional to the gradient of the log-likelihood of the target distribution [131, 174, 175], the noise estimated by an unconditional diffusion model has been shown to serve as a valuable prior for regularisation during training [82]. In Chapter 7, we adopt a similar technique for 3D data by training an unconditional point cloud diffusion model specifically tailored to indoor scenes. This model is then used as a regulariser across various scene coordinate regression tasks during training. Our results show that this regularisation approach effectively reduces noise in the reconstructed 3D scenes, resulting in more accurate geometry and improved performance in downstream tasks.

Chapter 3

Ray-ONet: Efficient 3D Reconstruction From A Single RGB Image

Reconstructing a 3D object from a single image is a highly challenging problem due to inherent ambiguities in scale and occluded regions. Learning shape priors from large datasets can assist in hallucinating the complete geometry; however, achieving a balance between capturing a reasonable global structure and preserving fine-grained details remains difficult.

For representing object geometry, explicit representations such as voxels or meshes are limited by resolution constraints, often lacking fine details and requiring large model sizes. In contrast, implicit representations, such as Occupancy Networks(ONet) [176] and DeepSDF [25], offer higher fidelity but at the cost of significantly increased inference time.

In this chapter, we present Ray-ONet, a model with high expressiveness for encoding object-level geometric prior for monocular reconstruction while maintaining efficiency during inference. We adopt a hybrid ray-based representation, where,

for each camera ray passing through an image pixel, we predict occupancies along the ray. Compared to ONet, which require querying arbitrary 3D points, the ray-based querying strategy reduces the inference time complexity while maintaining a similar model size. Our model learns multi-level, ray-based image features that capture both global categorical context and local fine-grained information. This design leads to improved reconstruction performance and strong generalisation ability to unseen object categories and real-world images.

To summarise, our main contributions are as follows:

- We introduce a novel ray-based 3D representation that enables more efficient inference while maintaining a similar memory footprint to ONet [176].
- We design a model that encodes geometric priors using multi-scale image features, capturing both global and local information as priors for high-fidelity reconstruction.
- We demonstrate state-of-the-art performance on ShapeNet [177] and strong generalisation ability to objects from unseen categories and real-world images.

Below, we introduce our paper *Ray-ONet: Efficient 3D Reconstruction From A Single RGB Image* presented at *British Machine Vision Conference (BMVC) 2021*.

Ray-ONet: Efficient 3D Reconstruction From A Single RGB Image

Wenjing Bian Zirui Wang Kejie Li Victor Adrian Prisacariu
Active Vision Lab, University of Oxford
{wenjing, ryan, kejie, victor}@robots.ox.ac.uk

Abstract

We propose Ray-ONet to reconstruct detailed 3D models from monocular images efficiently. By predicting a series of occupancy probabilities along a ray that is back-projected from a pixel in the camera coordinate, our method Ray-ONet improves the reconstruction accuracy in comparison with Occupancy Networks (ONet), while reducing the network inference complexity to O(N^2). As a result, Ray-ONet achieves state-of-the-art performance on the ShapeNet benchmark with more than 20× speed-up at 128^3 resolution and maintains a similar memory footprint during inference.

1. Introduction

3D reconstruction from a single view is an inherently ill-posed problem. However, it is trivial for a human to hallucinate a full 3D shape of a chair from a single image, since we have seen many different types of chairs in many angles in our daily life. Therefore, recovering a plausible 3D geometry from a single view is possible when enough prior knowledge is available, and in certain situations, might be the only feasible method when multi-view constrains are inaccessible, for instance, creating a 3D chair model from a chair image on a random website for an Augmented Reality (AR) application.

Recently, the single-view object reconstruction field has been advanced significantly using deep learning. In particular, deep implicit representations, such as DeepSDF [19], Occupancy Networks [17] and their extensions [20, 24], demonstrate promising results on the single-view 3D reconstruction task, by representing 3D shapes via a continuous function that is embedded in network parameters. Although these methods enjoy the advantages of representing 3D models in any resolutions and require small memory footprint and low storage space, the inference speed is still an issue yet to address, as most existing methods require querying millions of 3D locations to recover one 3D shape, leading to an O(N^3) complexity when a simple 3D sampling strategy is employed.

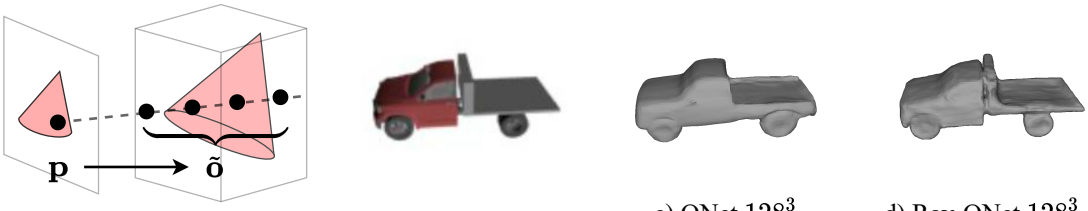
In this paper, we propose to predict a series of occupancies over 3D positions along a ray back-projected from a query 2D location on an image plane given local and global image features jointly. This formulation – view-centered ray-based occupancy prediction, taking advantage of local and global image features – improves the reconstruction accuracy while reducing the complexity to O(N^2).

One issue of this viewer-centered reconstruction is the depth/scale ambiguity presented by a single view (*i.e.*, the same image can be rendered by a small object that is close to the camera or a large object that is at a distance). A common practice is to normalise all ground-truth shapes into a constant scale during training. However, this operation implicitly entangles the shape learning with the scale ambiguity. Instead, we show how one can decouple these two problems and improve the reconstruction accuracy by providing the network with a scale calibration factor, which consists of a camera focal length and a camera-object distance.

In summary, we introduce Ray-ONet with three contributions: First, our method predicts a series of occupancies over 3D positions along a ray from a 2D query point, by conditioning a occupancy prediction function to both global and local image context, reducing the complexity to O(N^2). Second, we introduce a scale calibration factor, to compensate the issue caused by different object scales. Third, we demonstrate on ShapeNet that our method improves the overall 3D reconstruction accuracy while being 20× faster than ONet.

2. Related Work

Shape Representation. Computation efficiency and reconstruction quality are both effected by the choice of representation for the model output. Researchers resort to volumetric representations [5, 32] when deep networks are used for shape generation because 3D voxel grids naturally fit into the convolutional operation. However, the resolution of voxels is often limited by memory, which grows exponentially for high-resolution outputs. Alternatively, point clouds have been adopted [7], but they cannot represent surfaces directly. Meshes [30], being compact and surface aware, assume a pre-defined topology.



a) Ray-ONet b) Input image c) ONet 128^3 **0.333s** per object d) Ray-ONet 128^3 **0.014s** per object

Figure 1. Ray-ONet recovers more details than ONet [17] while being 20× faster, by predicting a series of occupancies $\tilde{\mathbf{o}}$ along the ray that is back-projected from a 2D point $\mathbf{p}$.

In recent years, implicit functions [4, 17, 19] are widely used for representing continuous 3D shapes. Networks using these implicit representations are trained such that the decision boundary represents the iso-surface of an object shape. These models are able to reconstruct high resolution 3D shapes with low memory cost, but the inference time increases cubically with sampling resolution. Hybrid models [22, 23] which combine two representations have also been developed recently. They couple explicit and implicit representations with 2 branches, where the explicit branch accelerates surface reconstruction while the implicit branch recnostruct fine details of object shapes, but the network size is inevitably increased dramtically. In contrast, Ray-ONet is able to speed up inference without increasing the memory footprint by predicting occupancy probability along a ray in a single network forward.

Single-View 3D Reconstruction. Reconstructing from a single view is a severly ill-posed problem, which was traditionally solved by "shape-from-X" techniques [13, 33] (X could be silhouette, shading etc.) using geometric or photometric information. The resurgence of deep learning in recognition tasks has encouraged researchers to solve single-view 3D reconstruction in a data driven manner. PSGN [7], 3D-R2N2 [5], and girdhar2016learning [8] pioneer using a deep network to learn a mapping from an RGB image to an object shape represented by voxels or point clouds. Following works are able to achieve better performance by using more sophisiticated representations (e.g. Octree [28], dense point clouds [15], or implicit representations [17]) or building shape prior [31].

More recently, CoReNet [21] predicts translation equivariant voxel grid using a variable grid offset. It enables fast mesh extraction but still requires significantly more memory than implicit models. The closest to our work is SeqXY2SeqZ [10], which also tries to reduce the time complexity of inference in implicit representations. To this end, they design a hybrid representation, which descritises the 3D space in two axes while keeping the rest continuous. At inference, they use a RNN to sequentially predict the start and end locations of occupied segments along the continuous axis.

Our work is also a hybrid approach, which explicitly predicts occupancy probabilities along a certain ray that passes through a point on the input image. Unlike Han et al. [10], we predict occupancies along a ray in a single network forward, which results in further reduction on time complexity. In addition, the coordinate system of Ray-ONet is aligned with the input image to take advantage of local features. As a result, we can predict 3D shapes with fine details.

Coordinate System. Two types of coordinate systems are used in single-view object reconstruction: i) view-centered aligned to to the input image or; ii) object-centered aligned to a canonical position in the shape database. As discussed in [25, 29], an object-centered coordinate system with aligned objects of the same category encourages learning category information, which provides a robust prior for learnt categories, whereas predictions made in the view space rely more on geometric information from the input image, and can generalise better to unseen categories. Furthermore, recent works [21] using the viewer-centered coordinate system demonstrate reconstruction with fine details by using image local feature. However, they often assume 3D shapes are normalised to a constant scale. Instead, we introduce a scale calibration factor to facilitate training given GT shapes in different scales.

3. Preliminary

Occupancy network (ONet) [17] formulates shape reconstruction as a continuous function mapping problem, where the goal is to learn a function $\mathcal{F}$ that maps a 3D position $\mathbf{x} \in \mathbb{R}^3$ to an occupancy probability $\tilde{o}$, where $\{\tilde{o} \in \mathbb{R} : 0 < \tilde{o} < 1\}$. The object shape is implicitly represented by the decision boundary of the learned function $\mathcal{F}$. To capture shape variations, the occupancy function $\mathcal{F}$ above is often conditioned on an observation, for example, an input image. Mathematically, the occupancy prediction function $\mathcal{F}$ can be defined as:

$$\mathcal{F}(\mathbf{x}, \mathbf{z}) = \tilde{o}, \tag{1}$$

where $\mathbf{z}$ is a latent code that encodes an input image I.

This mapping function, however, suffers from slow inference speed because the network needs to query a large number of 3D positions to recover a high-resolution 3D reconstruction, which leads to O(N^3) complexity if the 3D positions are sampled in a regular 3D grid. Although more sophisticated sampling strategies such as Multi-resolution IsoSurface Extraction (MISE) can be applied as in ONet [17], the inference speed is still largely restrained by this query-every-3D-point strategy. This motivates us to develop a novel ray-based pipeline that reduces the complexity to O(N^2).

4. Method

Given an input image I, we aim to build a 3D reconstruction in the camera space of I, instead of in the canonical space as in ONet, by predicting series of occupancies along rays that are back-projected for all image pixels. In other words, for each ray points to a 2D position $\mathbf{p}$ on the image plane, we predict a series of occupancies along the ray behind the 2D position in one forward pass, hence reducing the complexity to O(N^2). As a ray direction can be uniquely defined by the 2D position $\mathbf{p} = (p_x, p_y)$ in the viewer space, our Ray-ONet, which conditions the occupancy prediction on the ray direction, can be considered as conditioned on the 2D position $\mathbf{p}$. Fig. 2 presents an overview of our method.

In the following sections, we first formalise our occupancy prediction function in Section 4.1, followed by how we assemble it with image encoders and fusion modules (Section 4.2) and an detail description of the scale calibration factor that facilitates training given various scales of ground-truth object shapes (Section 4.3). Lastly, we describe our training and inference pipeline in Section 4.4 and Section 4.5, including the training loss and the mesh extracting process etc.

4.1. Occupancy Prediction

We now detail our occupancy prediction function. In Ray-ONet, we propose a function $\mathcal{G}$:

$$\mathcal{G}(\overbrace{\mathbf{p}, C_{\mathbf{p}}}^{\text{Local}}, \overbrace{\mathbf{z}, s}^{\text{Global}}) = \tilde{\mathbf{o}}, \tag{2}$$

which maps a 2D position $\mathbf{p}$ on an image I to a number of M occupancy estimations $\tilde{\mathbf{o}} \in \mathbb{R}^M$, conditioning on a global image feature $\mathbf{z}$, a local image feature $C_{\mathbf{p}}$ and a scale calibration factor s. Technically, the occupancy prediction function $\mathcal{G}$ is parameterised by a Multi-Layer Perceptron (MLP), and image global features $\mathbf{z}$ and local features $C_{\mathbf{p}}$ are encoded by a shared CNN encoder. From now on, with a slight abuse of notation, we also refer $\mathcal{G}$ as our ray occupancy network, as the network is just a parameterisation of the occupancy prediction function.

Comparing with the $\mathcal{F}(\mathbf{x}, \mathbf{z})$ in the classical ONet in Eq. (1), which maps a 3D position $\mathbf{x}$ to an occupancy estimation $\tilde{o}$ while conditioning on a global image feature, Ray-ONet uses $\mathcal{G}(\mathbf{p}, C_{\mathbf{p}}, \mathbf{z}, s)$ to map a 2D position $\mathbf{p}$ to a series of occupancy estimations $\tilde{\mathbf{o}}$ by conditioning the occupancy function on both local and global context. Ray-ONet, therefore, can predict all M occupancies along a ray in a single network forward pass, reducing the complexity to O(N^2). Moreover, as both our occupancy prediction function $\mathcal{G}$ and the classical $\mathcal{F}$ in ONet are parameterised in a similar MLP structure, our method maintains a similar memory footprint during inference.

4.2. Individual Modules

Given an input image I and its intrinsic and extrinsic camera parameters, we aim to extract relevant information for the occupancy prediction function $\mathcal{G}$. Specifically, we split the occupancy prediction function $\mathcal{G}$ to three modules: First, an image encoder $\mathcal{H}$, which extracts image global feature $\mathbf{z}$ and local feature $C_{\mathbf{p}}$; Second, a local context mixer $\mathcal{J}$, which fuses local image feature $C_{\mathbf{p}}$ with a ray that is defined by the 2D position $\mathbf{p}$ that it points to; and Third, an occupancy decoder $\mathcal{D}$, which decodes a series of occupancy estimations from both local and global context.

Image Encoder $\mathcal{H}$. We extract a global feature vector $\mathbf{z}$ and a feature map C from the input image I using a ResNet-fashion CNN [11] $\mathcal{H} : I \rightarrow (\mathbf{z}, C)$. Specifically, the global latent code $\mathbf{z}$ is an output of a ResNet-18, and the feature map $C \in \mathbb{R}^{H \times W \times D}$ is a result of concatenating bilinear upsampled activations after the ResNet blocks, where H and W denote input image resolution, and D denotes the feature dimension.

Local Context Mixer $\mathcal{J}$. We parameterise the local context mixer module with an MLP $\mathcal{J} : (C_{\mathbf{p}}, \mathbf{p}) \rightarrow \mathbf{y}_{\mathbf{p}}$, which aggregates a 2D location $\mathbf{p}$ and an interpolated image feature $C_{\mathbf{p}}$ to a fused local context feature $\mathbf{y}_{\mathbf{p}}$.

Occupancy Decoder $\mathcal{D}$. For a local feature $\mathbf{y}_{\mathbf{p}}$ at point $\mathbf{p}$, we aim to predict a series of occupancies, conditioning on a global feature $\mathbf{z}$ and a scale calibration factor s, via a decoder $\mathcal{D} : (\mathbf{y}_{\mathbf{p}}, \mathbf{z}, s) \rightarrow \tilde{\mathbf{o}}$. Similar to ONet, we parameterise the decoder $\mathcal{D}$ in an MLP network with skip links and Conditional Batch Normalisation (CBN).

To summaries this section, we assemble the occupancy prediction network $\mathcal{G}$ with three modules: an image encoder $\mathcal{H}$, a local context mixer $\mathcal{J}$ and an occupancy decoder $\mathcal{D}$:

$$\mathcal{G}(\overbrace{\mathbf{p}, C_{\mathbf{p}}}^{\text{Local}}, \overbrace{\mathbf{z}, s}^{\text{Global}}) = \mathcal{D}(\mathcal{J}(\mathbf{p}, C_{\mathbf{p}}), \mathbf{z}, s) = \tilde{\mathbf{o}}, \tag{3}$$

where the global image feature $\mathbf{z}$ and the image feature map C are extracted the input image I from encoder $\mathcal{H}$.

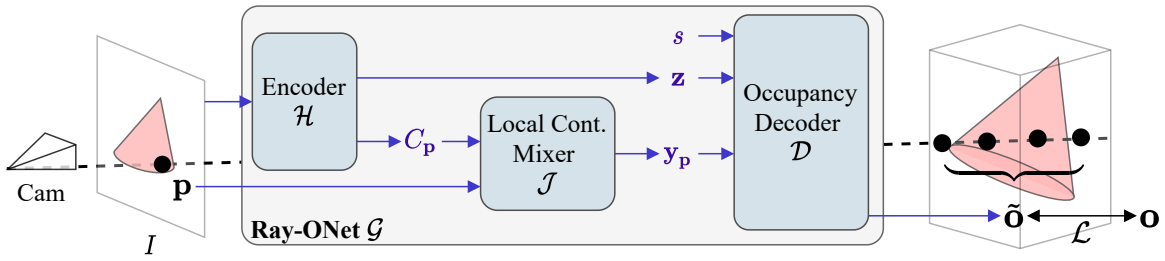


Figure 2. **Ray-ONet Overview.** Given an RGB image, Ray-ONet first maps the image to a global feature descriptor $\mathbf{z}$ and a dense feature map C via an image encoder $\mathcal{H}$. The local feature $C_\mathbf{p}$ (feature at position $\mathbf{p}$) and the position $\mathbf{p}$, taken by the local context mixer $\mathcal{J}$, are mapped to a local context descriptor $\mathbf{y_p}$. Finally, the occupancy decoder $\mathcal{D}$ takes as input $\mathbf{y_p}$, $\mathbf{z}$ and a scale factor s and predicts occupancies of 3D points along the back-projected ray from position $\mathbf{p}$.

4.3. Scale Calibration

The scale of the predictions is as important as the reconstructed shape because a correct shape prediction but in a different scale to the ground-truth can result in a large training loss. Instead of normalising all ground-truth shapes, we introduce the scale calibration factor s as an extra dimension for global feature to facilitate network training given ground-truth shapes in different scales. Specifically, we define our scale calibration factor $s = \|\mathbf{c}\|_2/f$, where $\|\mathbf{c}\|_2$ denotes the distance between a camera and a ground truth 3D model, and f denotes the camera focal length.

This scale calibration factor s conveys that the predicted shape scale should be proportional to camera-object distance and inverse proportional to the camera focal length. In the case of using a fixed focal length for the entire training set, the scale calibration factor can be further reduced to $\|\mathbf{c}\|_2$. Note that this scale calibration factor is only needed during training.

4.4. Training

We use a binary cross entropy loss $\mathcal{L}$ between predicted occupancies $\tilde{\mathbf{o}}$ and ground truth occupancies $\mathbf{o}$ to supervise the training of the encoder $\mathcal{H}$, the mixer $\mathcal{J}$ and the decoder $\mathcal{D}$:

$$\mathcal{L}(\tilde{\mathbf{o}}, \mathbf{o}) = -[\mathbf{o}\log(\tilde{\mathbf{o}}) + (1-\mathbf{o})\log(1-\tilde{\mathbf{o}})]. \quad (4)$$

As we predict 3D models in the camera space of an input image, we rotate ground truth 3D models so that the projection of a ground truth 3D model on the input image plane matches the observations on I. This step is necessary to acquire ground truth occupancies $\mathbf{o}$ in the camera space of an input image.

4.5. Inference

During inference, as we do not have access to camera-object distance, nor camera intrinsics, the scale calibration factor can be set to a random value in the range of the scaling factor used during training. As a result, our method recovers correct shapes in an unknown scale due to the inherent scale/depth ambiguity in the single-view 3D reconstruction task.

Before extracting meshes using the Marching Cubes algorithm [16], we apply an additional re-sampling process on the Ray-ONet occupancy predictions, which is in the shape of a camera frustum due to the nature of our ray-based method, to get occupancies in a regular 3D grid. This re-sampling process can be simply done by a trilinear interpolation.

5. Experiments

We conduct experiments to evaluate our method on two benchmarks, showing our method achieves state-of-the-art performance while being much faster during inference. This section is organised as follows, Section 5.1 reports our experiments setup. We compare our reconstruction quality with baselines in Section 5.2. Memory and time consumption are reported in Section 5.3. Lastly, we conduct an ablation study in Section 5.4.

5.1. Setup

Dataset. We evaluate Ray-ONet on the ShapeNet dataset [2], which is a large-scale dataset containing 3D synthetic object meshes with category labels. To make fair comparisons with prior works, we follow the evaluation protocol in ONet [17], taking a pre-processed subset from 3D-R2N2 [5], which includes image renderings from 13 classes in ShapeNetCore v1 [2] and adopting the same train/eval/test splits.

Metrics. Following ONet [17], we adopt the volumetric IoU, the Chamfer-L1 distance, and the normal consistency score (NC) as the evaluation metrics.

Implementation Details. Network initialisation: Apart from the image encoder $\mathcal{H}$, which is fine tuned from an ImageNet [6]-pretrained ResNet-18 backbone [11] with a random initialised fully-connected layer head, the local con-

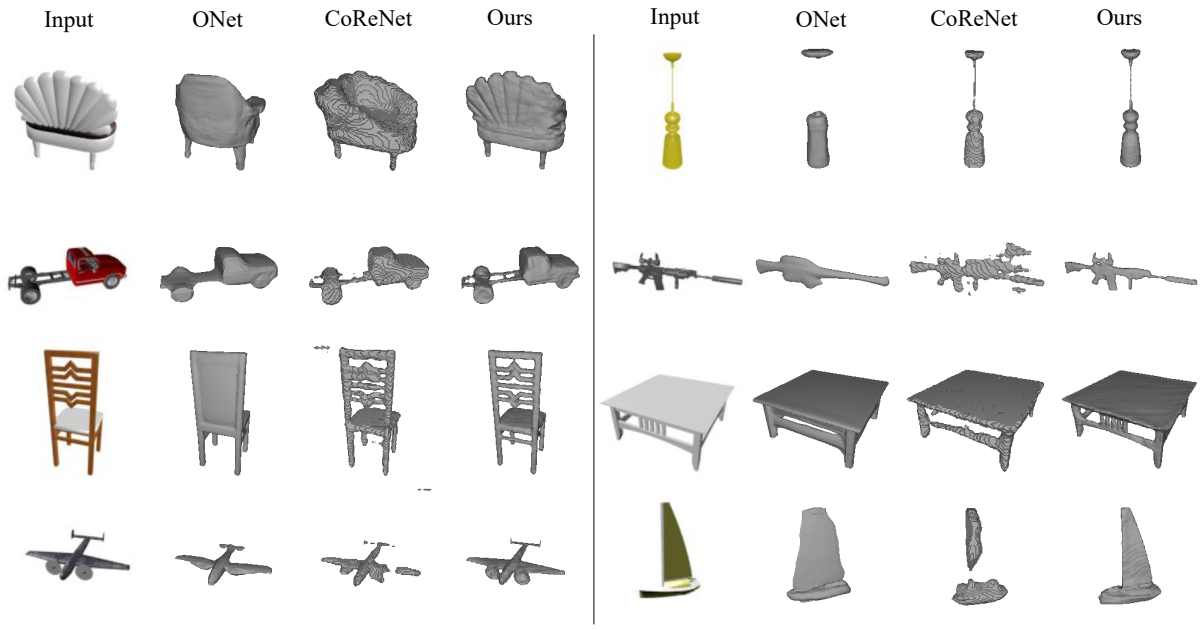


Figure 3. **Qualitative Reconstruction Results on ShapeNet.** Note that Ray-ONet is able to reconstruct fine details of object shapes, such as the complicated pattern on the chair back.

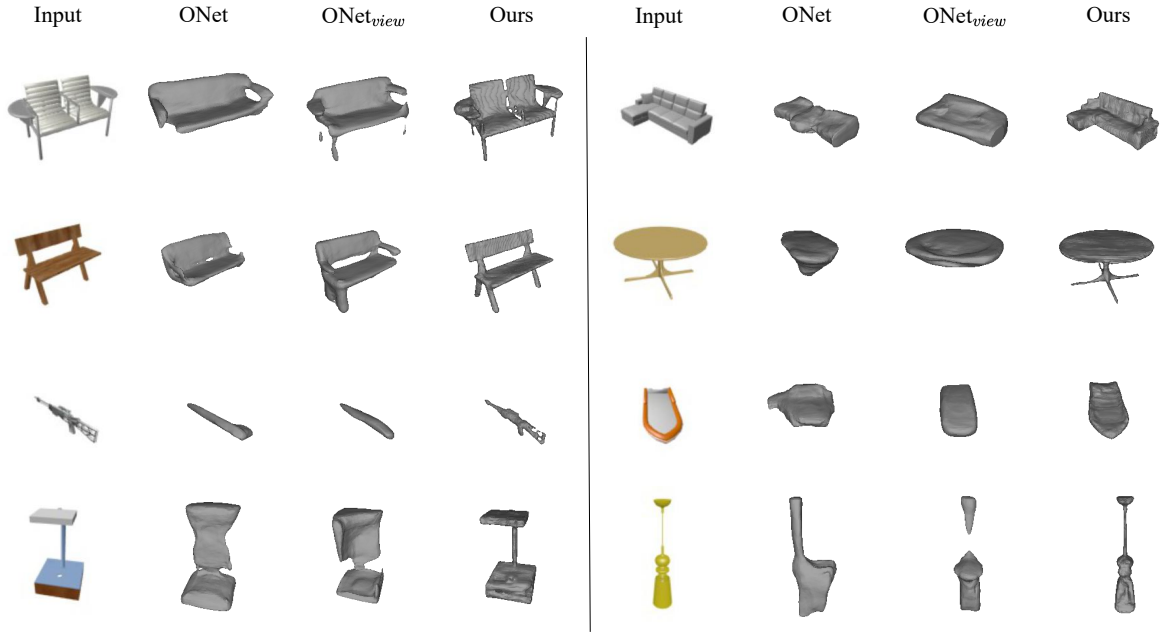


Figure 4. **Qualitative Reconstruction Results on Unseen Categories.** We show Ray-ONet can generalise better than our baselines on unseen categories, in which object shapes are significantly different from the training classes (airplane, car, and chair).

text mixer $\mathcal{J}$ and the occupancy decoder $\mathcal{D}$ are both trained from scratch. Training details: During training, we randomly sample 1024 pixels on an input image and predict occupancies for 128 points that are equally spaced between a defined depth range on the rays. We use a batch size of 64, with an Adam [12] optimiser and a learning rate of 10^{-4}.

Network architecture: A detailed description of network architectures is available in the supplementary material.

5.2. Reconstruction Accuracy

Standard Reconstruction. We evaluate the performance of Ray-ONet on the standard 3D reconstruction task, train-

ing/testing on all 13 categories. We compare our model with prior works including ONet [17], AtlasNet [9], CoReNet [21], DmifNet [14], and 3D43D [1]. Note that ONet, AtlasNet and DmifNet are object-centered models and the others are view-centered. To make a fair comparison, we additionally train an ONet model with the view-centered setup, named ONet$_{view}$. Table 1 reports that our model achieves state-of-the-art 3D reconstruction accuracy on the ShapeNet benchmark.

Reconstruction on Unseen Categories. To evaluate the generalisation capability, we train our model on 3 classes (airplane, car, and chair) and test on the remaining 10 classes. Table 2 reports that our model performs better than all baselines in this setup.

5.3. Memory and Inference Time

As shown in Table 3, Ray-ONet is over 20× faster than ONet even if the efficient Multi-resolution IsoSurface Extraction is used in ONet while maintaining similar memory footprint. It is worth noting that 3D-R2N2 [5] uses a 3D auto-encoder architecture to predict on discretised voxel grid. Although it is 4× faster than Ray-ONet, it produces inferior reconstruction accuracy and requires much larger memory size with 20× parameters. Since SeqXY2SeqZ [10] has not released their code, we can only compare with them on complexity and accuracy. The inference time complexity of SeqXY2SeqZ is $O(N^2) \times l_{RNN}$, where l_{RNN} denotes the number of the recurrent steps. While being more efficient, we outperform SeqXY2SeqZ by 0.06 measured in IoU at 32^3 resolution (SeqXY2SeqZ reports its IoU at this resolution).

5.4. Ablation Study

We analyse the effectiveness of the scale calibration factor s as well as the global and local image features.

Effect of the Scale Calibration Factor. Training with the scale calibration factor can lead to better reconstruction accuracy as reported in Table 4. This shows that decoupling scale from shape learning not only enables us training with varying scales of ground-truth shapes but also helps in network convergence.

Effect of Global and Local Image Features. Both global and local image features are important for reconstruction accuracy as suggested by the performance gap between w/ and w/o features in Table 4.

6. Conclusion

In this paper, we proposed Ray-ONet, a deep neural network that reconstructs detailed object shapes efficiently given a single RGB image. We demonstrate that predicting a series of occupancies along back-projected rays from both global and local features can significantly reduce the time complexity at inference while improving the reconstruction quality. Experiments show that Ray-ONet is able to achieve state-of-the-art performance for both seen and unseen categories in ShapeNet with a significant evaluation time reduction. As our single-view object reconstruction results are encouraging, we believe an interesting direction in the future is to extend Ray-ONet to tasks suchs as scene-level reconstructions or novel view synthesis.

Acknowledgements

We thank Theo Costain for helpful discussions and comments. We thank Stefan Popov for providing the code for CoReNet and guidance on training. Wenjing Bian is supported by China Scholarship Council (CSC).

References

[1] Miguel Angel Bautista, Walter Talbott, Shuangfei Zhai, Nitish Srivastava, and Joshua M Susskind. On the generalization of learning-based 3d reconstruction. In WACV, 2021. 6

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012, 2015. 4, 9

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. TPAMI, 2017. 9

[4] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In CVPR, 2019. 2

[5] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In ECCV, 2016. 1, 2, 4, 6, 9

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, 2009. 4, 9

[7] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In CVPR, 2017. 1, 2

[8] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In ECCV, 2016. 2

[9] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In CVPR, 2018. 6

IoU ↑	airplane	bench	cabinet	car	chair	display	lamp	loudspeaker	rifle	sofa	table	telephone	vessel	mean
ONet	0.591	0.492	0.750	0.746	0.530	0.518	0.400	0.677	0.480	0.693	0.542	0.746	0.547	0.593
DmifNet	**0.603**	0.512	0.753	**0.758**	0.542	0.560	0.416	0.675	0.493	0.701	0.550	0.750	0.574	0.607
ONet_{view}	0.418	0.411	0.737	0.693	0.480	0.526	0.374	0.685	0.348	0.659	0.520	0.694	0.456	0.538
CoReNet†	0.411	0.380	0.660	0.664	0.464	0.470	0.349	0.669	0.407	0.628	0.469	0.616	0.464	0.512
3D43D	0.571	0.502	0.761	0.741	0.564	0.548	0.453	**0.729**	0.529	0.718	0.574	0.740	0.588	0.621
Ours	0.574	**0.521**	**0.763**	0.755	**0.567**	**0.601**	**0.467**	0.726	**0.594**	**0.726**	**0.578**	**0.757**	**0.598**	**0.633**
Chamfer-L1 ↓	airplane	bench	cabinet	car	chair	display	lamp	loudspeaker	rifle	sofa	table	telephone	vessel	mean
AtlasNet*	0.104	0.138	0.175	0.141	0.209	0.198	0.305	0.245	0.115	0.177	0.190	0.128	**0.151**	0.175
ONet	0.134	0.150	0.153	0.149	0.206	0.258	0.368	0.266	0.143	0.181	0.182	0.127	0.201	0.194
DmifNet	0.131	0.141	0.149	0.142	0.203	0.220	0.351	0.263	0.135	0.181	0.173	0.124	0.189	0.185
ONet_{view}	0.247	0.188	0.163	0.193	0.256	0.242	0.820	0.248	0.219	0.205	0.181	0.151	0.280	0.261
3D43D	**0.096**	**0.112**	**0.119**	**0.122**	0.193	**0.166**	0.561	0.229	0.248	**0.125**	0.146	**0.107**	0.175	0.184
Ours	0.121	0.124	0.140	0.145	**0.178**	0.183	**0.238**	**0.204**	**0.094**	0.151	**0.140**	0.109	0.163	**0.153**
NC ↑	airplane	bench	cabinet	car	chair	display	lamp	loudspeaker	rifle	sofa	table	telephone	vessel	mean
AtlasNet*	0.836	0.779	0.850	0.836	0.791	0.858	0.694	0.825	0.725	0.840	0.832	0.923	0.756	0.811
ONet	0.845	0.814	0.884	0.852	0.829	0.857	0.751	0.842	0.783	0.867	0.860	**0.939**	0.797	0.840
DmifNet	**0.853**	0.821	0.885	**0.857**	0.835	0.872	0.758	0.847	0.781	0.873	0.868	0.936	0.808	0.846
ONet_{view}	0.785	0.798	0.884	0.838	0.816	0.875	0.731	0.861	0.732	0.863	0.865	0.932	0.766	0.827
3D43D	0.825	0.809	**0.886**	0.844	0.832	**0.883**	0.766	**0.868**	0.798	0.875	0.864	0.935	0.799	0.845
Ours	0.830	**0.828**	0.881	0.849	**0.844**	**0.883**	**0.780**	0.866	**0.837**	**0.881**	**0.873**	0.929	**0.815**	**0.854**

Table 1. **Reconstruction Accuracy on ShapeNet.** We show Ray-ONet achieves state-of-the-art reconstruction accuracy in all three evaluation metrics. Notation * denotes results taken from [17] and † denotes results we train their public code and evaluate in ONet's protocol. All other results are taken from corresponding original publications.

IoU ↑	bench	cabinet	display	lamp	loudspeaker	rifle	sofa	table	telephone	vessel	mean
ONet	0.251	0.400	0.165	0.147	0.394	0.172	0.505	0.222	0.147	0.374	0.278
ONet_{view}	0.241	0.442	0.204	0.135	0.449	0.104	0.524	0.264	0.239	0.302	0.290
3D43D	0.302	0.502	0.243	0.223	0.507	0.236	0.559	0.313	0.271	**0.401**	0.356
Ours	**0.334**	**0.517**	**0.256**	**0.246**	**0.538**	**0.253**	**0.583**	**0.350**	**0.277**	0.394	**0.375**
Chamfer-L1 ↓	bench	cabinet	display	lamp	loudspeaker	rifle	sofa	table	telephone	vessel	mean
ONet	0.515	0.504	1.306	1.775	0.635	0.913	0.412	0.629	0.659	0.392	0.774
ONet_{view}	0.443	0.418	0.840	2.976	0.496	1.188	0.343	0.470	0.532	0.469	0.818
3D43D	0.357	0.529	1.389	1.997	0.744	0.707	0.421	0.583	0.996	0.521	0.824
Ours	**0.228**	**0.351**	**0.592**	**0.485**	**0.389**	**0.307**	**0.266**	**0.299**	**0.453**	**0.304**	**0.367**
NC ↑	bench	cabinet	display	lamp	loudspeaker	rifle	sofa	table	telephone	vessel	mean
ONet	0.694	0.697	0.527	0.563	0.693	0.583	0.779	0.701	0.605	0.694	0.654
ONet_{view}	0.717	**0.762**	0.600	0.508	0.740	0.505	0.796	0.740	0.673	0.679	0.672
3D43D	0.706	0.759	0.638	0.618	0.749	0.588	0.784	0.731	**0.700**	0.690	0.696
Ours	**0.736**	**0.762**	**0.640**	**0.642**	**0.762**	**0.609**	**0.806**	**0.764**	0.670	**0.703**	**0.709**

Table 2. **Reconstruction Accuracy on Unseen Categories.** We report the performance of different models on 10 unseen categories after trained on 3 categories. Our method outperforms all prior works in almost all categories.

Methods	ONet	3D-R2N2	CoReNet	Ours
# Params (M)	13.4	282.0	36.1	14.2
Inference Memory (MB)	1449	5113	2506	1559
Inference time (ms)	333.1	3.4	37.8	13.8

Table 3. **Comparisons of Number of Parameters, Inference Memory and Inference Time at 128^3 Resolution**. The inference time measures the network forward time only and does not include any post-processing time.

	IoU ↑	Chamfer-L_1 ↓	NC ↑
Ours	**0.633**	**0.153**	**0.854**
w/o scale s	0.545	0.226	0.818
w/o global $\mathbf{z}$	0.574	0.206	0.821
w/o local $C_{\mathbf{p}}$	0.525	0.224	0.802

Table 4. **Ablation Study.** We show the effectiveness of the scale calibration factor s, the global image feature $\mathbf{z}$ and the local image feature $C_{\mathbf{p}}$.

[10] Zhizhong Han, Guanhui Qiao, Yu-Shen Liu, and Matthias Zwicker. Seqxy2seqz: Structure learning for 3d shapes by sequentially predicting 1d occupancy segments from 2d coordinates. In ECCV, 2020. 2, 6

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian

Sun. Deep residual learning for image recognition. In CVPR, 2016. 3, 4, 9, 10

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In ICLR, 2015. 5

[13] Aldo Laurentini. The visual hull concept for silhouette-based image understanding. TPAMI, 1994. 2

[14] Lei Li and Suping Wu. Dmifnet: 3d shape reconstruction based on dynamic multi-branch information fusion. In International Conference on Pattern Recognition (ICPR), 2021. 6

[15] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In AAAI, 2018. 2

[16] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. ACM SIGGRAPH Computer Graphics, 1987. 4

[17] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In CVPR, 2019. 1, 2, 3, 4, 6, 7, 9

[18] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In CVPR, 2016. 9

[19] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In CVPR, 2019. 1, 2

[20] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In ECCV, 2020. 1

[21] Stefan Popov, Pablo Bauszat, and Vittorio Ferrari. Corenet: Coherent 3d scene reconstruction from a single rgb image. In ECCV, 2020. 2, 6

[22] Omid Poursaeed, Matthew Fisher, Noam Aigerman, and Vladimir G Kim. Coupling explicit and implicit surface representations for generative 3d modeling. In ECCV, 2020. 2

[23] Martin Runz, Kejie Li, Meng Tang, Lingni Ma, Chen Kong, Tanner Schmidt, Ian Reid, Lourdes Agapito, Julian Straub, Steven Lovegrove, et al. Frodo: From detections to 3d objects. In CVPR, 2020. 2

[24] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In CVPR, 2019. 1

[25] Daeyun Shin, Charless C Fowlkes, and Derek Hoiem. Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction. In CVPR, 2018. 2

[26] David Stutz and Andreas Geiger. Learning 3d shape completion from laser scan data with weak supervision. In CVPR, 2018. 9

[27] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In CVPR, 2018. 9

[28] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In ICCV, 2017. 2

[29] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In CVPR, 2019. 2

[30] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In ECCV, 2018. 1

[31] Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T Freeman, and Joshua B Tenenbaum. Learning shape priors for single-view 3d completion and reconstruction. In ECCV, 2018. 2

[32] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In ICCV, 2019. 1

[33] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading: a survey. TPAMI, 1999. 2

Appendix A. Implementation Details

The following sections include a detailed description of our model architecture, data preprocessing steps and evaluation procedure. We will release our code later.

Encoder $\mathcal{H}$. The encoder is built on a ResNet-18 architecture [11] with an additional upsampling step to generate feature maps. The encoder is initialised with pretrained weights on ImageNet dataset [6] except the last fully connected layer. The global feature $\mathbf{z}$ is obtained from a fully connected layer with D_{global} dimensions. The outputs from the 2^{nd}, 4^{th} and 6^{th} ResNet Blocks, are upsampled to 112×112 using bilinear interpolation and concatenated, togenether with the the 112×112 feature maps output from the 'Conv64' layer, to form 512 dimensional feature maps. The dimension is changed to D_{local} with a fully connected layer. The local feature is then extracted from the corresponding position of 2D point $\mathbf{p}$ on the image. In practice, we choose $D_{global} = D_{local} = 256$.

Local Context Mixer $\mathcal{J}$. As shown in Fig. 6, the Local Context Mixer $\mathcal{J}$ takes a batch of T local features with D_{local} dimensions and the corresponding 2D points as input. The local features and points are first concatenated and projected to 256 dimensions with a fully-connected layer. It then passes through 3 residual MLPs with ReLU activation before each fully-connected layer. The output local feature has T batches with 256 dimensions.

Occupancy Decoder $\mathcal{D}$. The Occupancy Decoder $\mathcal{D}$ follows the architecture of occupancy network [17], with different inputs and output dimensions. The inputs are the global feature $(\mathbf{z}, s)$ and the local feature output from $\mathcal{J}$. The local feature first passes through 5 pre-activation ResNet-blocks. Each ResNet-block consist of 2 sub-blocks, where each sub-block applies Conditional Batch-Normalization (CBN) to the local feature followed by a ReLU activation function and a fully-connected layer. The output from the ResNet-block is added to the input local feature. After the 5 ResNet-blocks, the output passes through a last CBN layer and ReLU activation and a final fully-connected layer which produces the M dimensional output, representing occupancy probability estimations for M points along the ray.

A.1. Data Preprocessing

We use the image renderings and train/test split of ShapeNet [2] as in 3D-R2N2 [5]. Following ONet [17], the training set is subdivided into a training set and a validation set.

In order to generate ground truth occupancies for rays from each view, we first create watertight meshes using the code provided by [26]. With camera parameters for 3D-R2N2 image renderings, we place the camera at corresponding location of each view, and generate 5000 random rays passing through the image. We sample equally spaced points on each ray between defined distances d_{min} and d_{max}. In practice, we choose $d_{min} = 0.63$ and $d_{max} = 2.16$, which guarantee all meshes are within this range.

A.2. Evaluation

To make a fair comparison with previous approaches, we use normalised ground truth points and point clouds produced by ONet [17] for evaluation. With the scale calibration factor, our predicted mesh has the same scale as the raw unnormalised ShapeNet mesh. In order to make the predicted mesh in a consistent scale as the normalised ground truth, we use the scale factor between the raw ShapeNet mesh and the normalised ShapeNet mesh to scale our prediction before evaluation. The threshold parameter for converting occupancy probabilities into binary occupancy values is set to 0.2 during inference.

Appendix B. Additional Experimental Results

B.1. Additional Qualitative Results on ShapeNet

Additional qualitative results on ShapeNet are shown in Fig. 7 and Fig. 8, where Fig. 7 is the standard reconstruction task results on categories seen during training, and Fig. 8 is the test results after trained on 3 different categories.

B.2. Qualitative Results on Online Products and Pix3D Datasets

With the model trained on 13 categories of synthesis ShapeNet [2] images, we made further tests on 2 additional datasets with real world images to validate the generalisation ability of the model.

Online Products Dataset. We use the chair category in Online Products dataset [18]. As the training data has white background, We first feed the image into DeepLabV3 [3] to generate a segmentation mask and changed the color of the image outside the mask to white. As there is no camera parameters available, the reconstruction shown in Fig. 9 is of correct proportion only.

Pix3D Dataset. Similarly, we test our model on chair category of Pix3D dataset [27], with the ground truth segmentation mask provided. Some results are shown in Fig. 10.

B.3. Limitations

As shown in Fig. 11, for certain objects in unseen categories, the 3D object reconstructions are much more accurate in the image view than other views, as our model is able to predict shape for visible parts from image features but lacks of shape priors for invisible parts on those unseen categories.

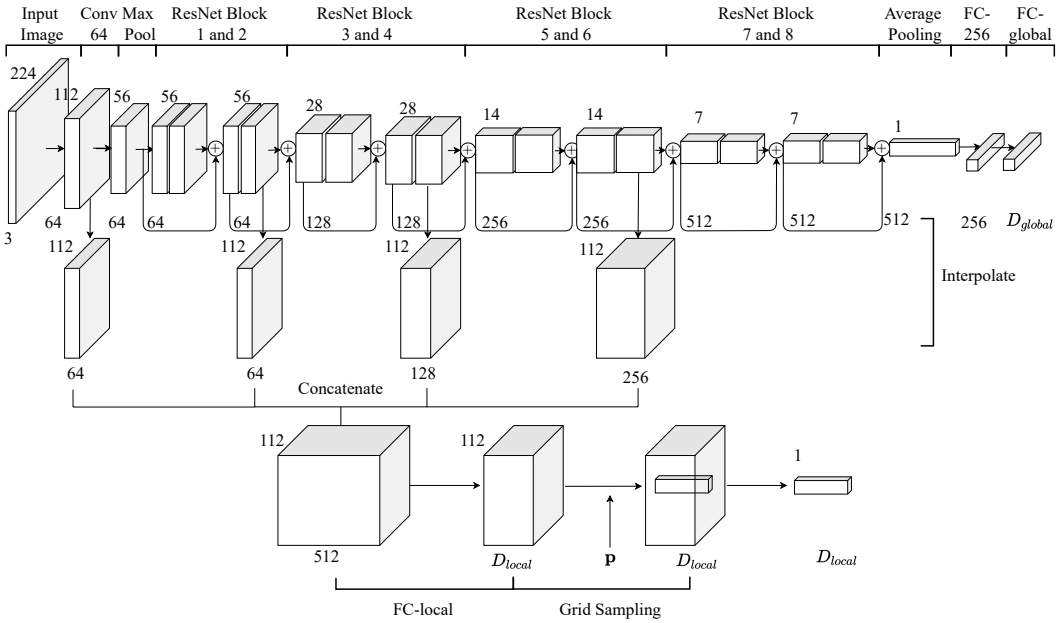


Figure 5. **Model Architecture for Encoder $\mathcal{H}$.** With a single image as input, the global latent code $\mathbf{z}$ is generated from a ResNet-18 [11]. The local image feature is extracted from the position of 2D location $\mathbf{p}$ on the concatenated feature maps from 4 different encoder stages. Symbol $\oplus$ represents additive operation.

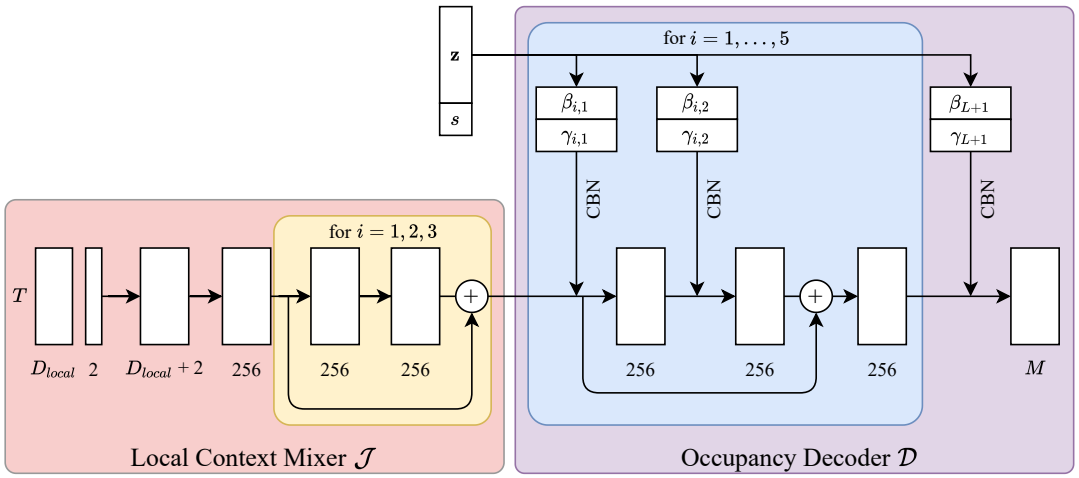


Figure 6. **Model Architecture for Local Context Mixer $\mathcal{J}$ and Occupancy Decoder $\mathcal{D}$.** Using the local feature and a 2D location $\mathbf{p}$ as input, we first generate a fused local context feature through Local Context Mixer $\mathcal{J}$. The Occupancy Decoder $\mathcal{D}$ uses Conditional Batch-Normalization (CBN) to condition the fused local feature on global latent code $\mathbf{z}$ and the scale calibration factor s, and predicts occupancy probabilities for M points along the ray.

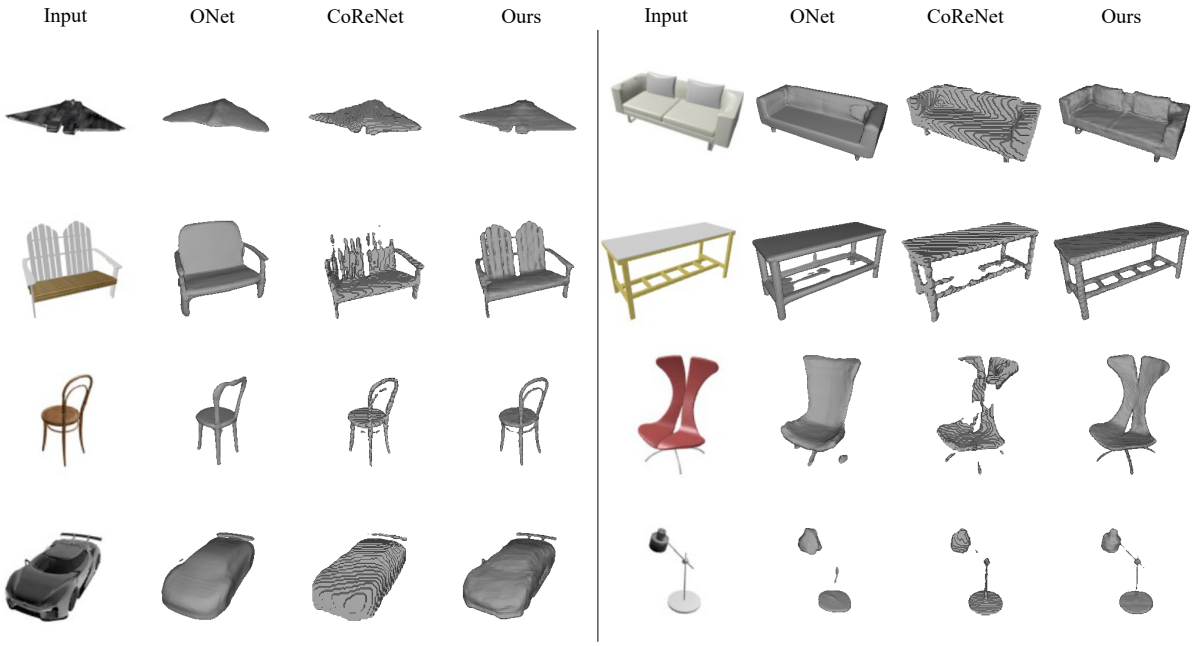


Figure 7. **Additional Qualitative Reconstruction Results on ShapeNet.**

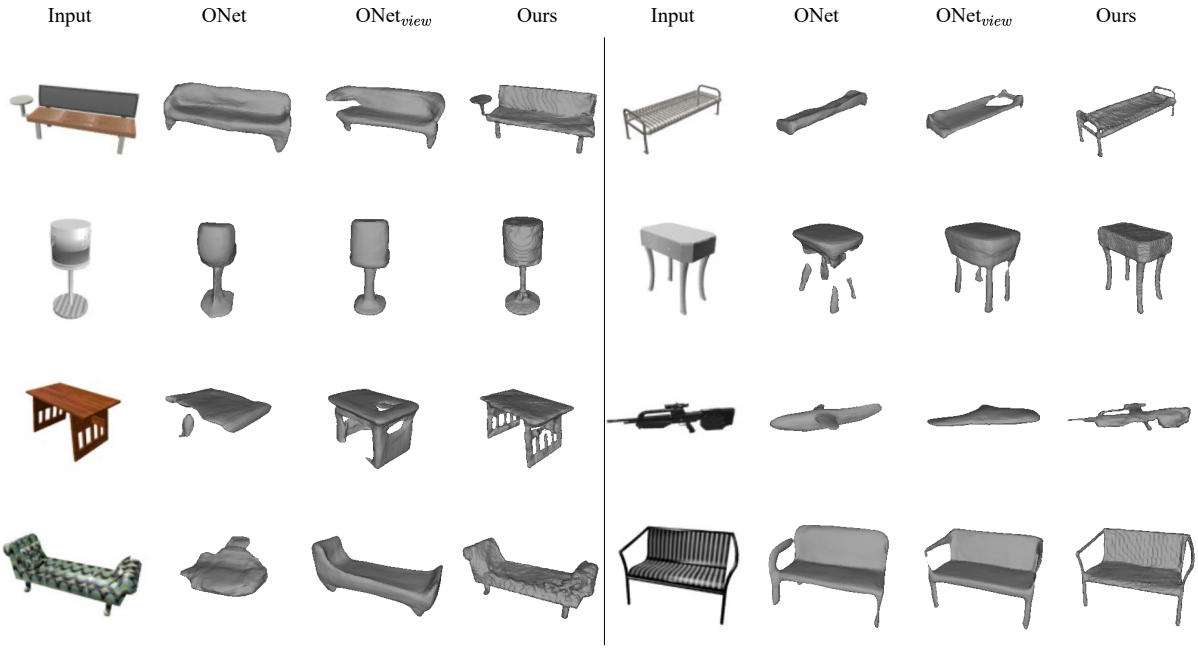


Figure 8. **Additional Qualitative Reconstruction Results on Unseen Categories.**

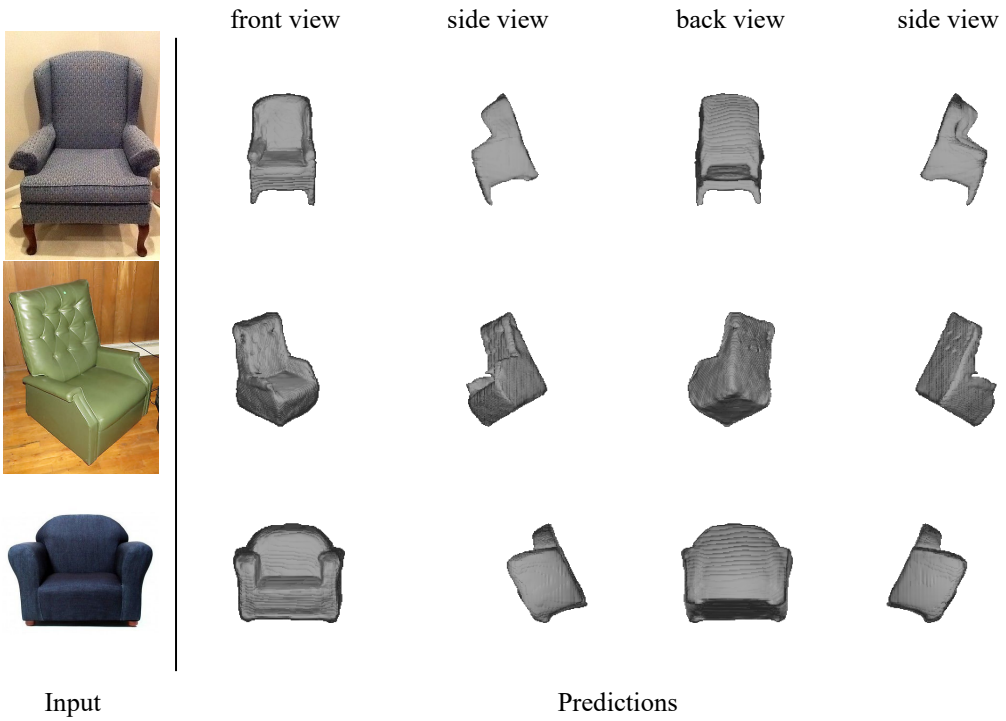


Figure 9. **Qualitative Reconstruction Results on Online Products dataset.**

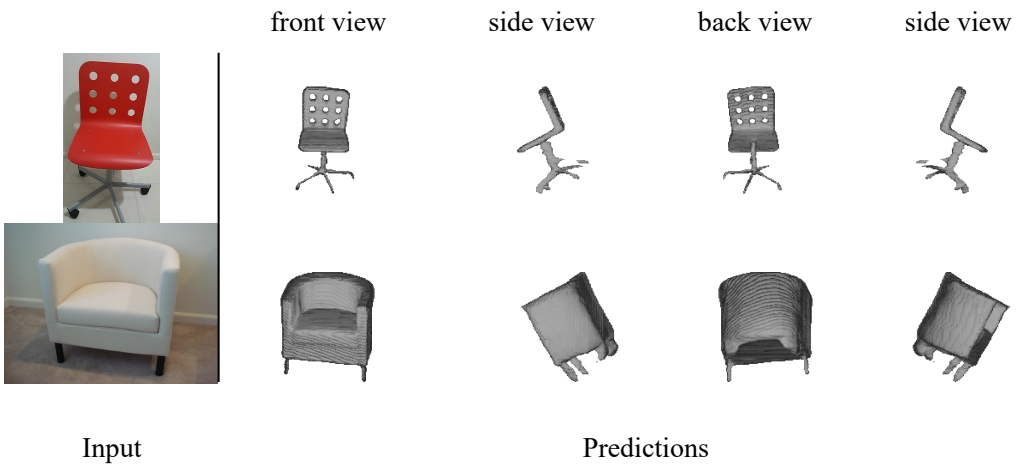


Figure 10. **Qualitative Reconstruction Results on Pix3D dataset.**

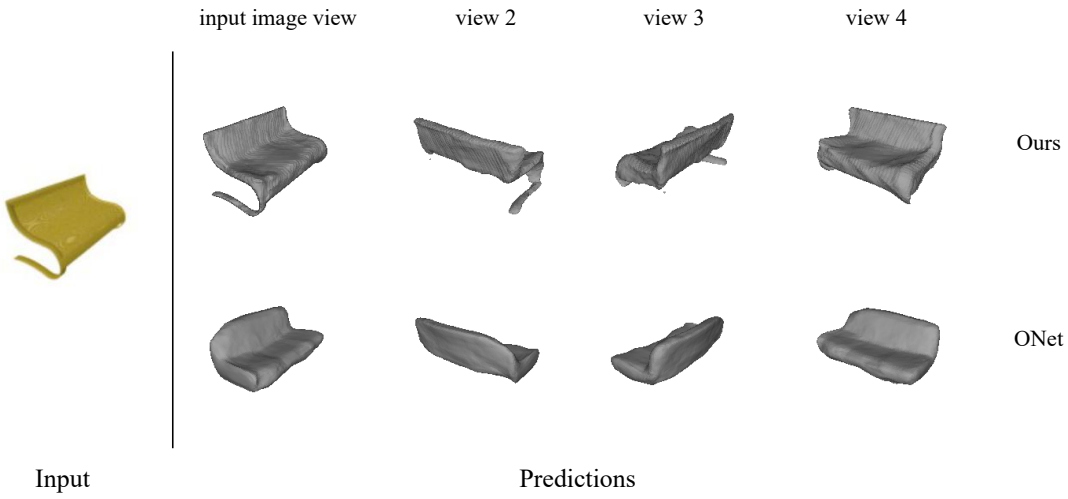


Figure 11. **A Failure Case for Novel Categories.**

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Ray-ONet: Efficient 3D Reconstruction From A Single RGB Image
Publication Status	☑ Published □ Accepted for Publication □ Submitted for Publication □ Unpublished and unsubmitted work written in a manuscript style
Publication Details	**Bian, W.**, Wang, Z., Li, K., & Prisacariu, V. A. (2021). Ray-ONet: efficient 3D reconstruction from a single RGB image. In *British Machine Vision Conference (BMVC)*.

Student Confirmation

Student Name:	Wenjing Bian		
Contribution to the Paper	• Conception and validation of the idea • Design of the architecture and framework • Implementation and running of all experiments • Research of prior work • Writing and presentation of the paper		
Signature	Wenjing Bian	Date	April 22, 2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Victor Adrian Prisacariu		
Supervisor comments The description is accurate		
Signature VPA	Date	April 22, 2025

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 4

CatFree3D: Category-agnostic 3D Object Detection with Diffusion

Detecting 3D objects from a single image is a longstanding problem in computer vision, with wide-ranging applications in augmented reality, virtual reality, and autonomous driving. Compared to 2D object detection, 3D detection requires the prediction of additional parameters such as object size, orientation, and depth. Moreover, the scale and diversity of 3D datasets lead most existing models to specialise in predefined categories and struggle to generalise to novel objects.

In this chapter, we present CatFree3D, a framework for detecting 3D objects without relying on category information during either training or inference. By introducing several visual priors including 2D bounding box and depth as model inputs, we enable a more general and flexible pipeline. Instead of directly regressing 3D bounding boxes, CatFree3D adopts a diffusion-based approach that denoises randomly initialised 3D boxes towards the target. This design eliminates the need for predefined region proposals or anchor boxes, allowing arbitrary number of predictions. To select the final predictions from multiple candidates, we also introduce a learnt confidence for each predicted box. CatFree3D achieves strong

performance on standard benchmarks and demonstrates remarkable generalisation to objects unseen during training and to in-the-wild images.

Additionally, we introduce a novel evaluation metric for 3D box accuracy, addressing limitations in existing metrics such as IoU and GIoU, and providing a more precise assessment of detection quality.

To summarise, our main contributions are:

- We propose a category-agnostic 3D object detection framework that leverages general visual priors, demonstrating strong generalisation to unseen objects.
- We introduce a diffusion-based 3D detection approach that removes the dependence on region proposals and enables multiple, confidence-scored predictions.
- We propose a new evaluation metric for 3D detection, offering a more precise and reliable measure of box accuracy.

The remainder of this chapter is based on our paper *CatFree3D: Category-agnostic 3D Object Detection with Diffusion* presented at *International Conference on 3D Vision (3DV) 2025*.

CatFree3D: Category-agnostic 3D Object Detection with Diffusion

Wenjing Bian[1] Zirui Wang[1] Andrea Vedaldi[2]
[1]Active Vision Lab [2]Visual Geometry Group
University of Oxford
{wenjing, ryan, vedaldi}@robots.ox.ac.uk
https://bianwenjing.github.io/CatFree3D

Abstract

Image-based 3D object detection is widely employed in applications such as autonomous vehicles and robotics, yet current systems struggle with generalisation due to complex problem setup and limited training data. We introduce a novel pipeline that decouples 3D detection from 2D detection and depth prediction, using a diffusion-based approach to improve accuracy and support category-agnostic detection. Additionally, we introduce the Normalised Hungarian Distance (NHD) metric for an accurate evaluation of 3D detection results, addressing the limitations of traditional IoU and GIoU metrics. Experimental results demonstrate that our method achieves state-of-the-art accuracy and strong generalisation across various object categories and datasets.

1. Introduction

Image-based 3D object detection systems are designed to identify and localise objects in three-dimensional space from input images. These systems play a critical role in applications like autonomous vehicles and robotics.

Recent developments in deep learning have significantly advanced 3D object detection [6, 11, 32, 39, 40, 51, 52, 69]. While these methods have been well engineered, they remain largely domain-specific and are limited in the number of detectable object categories, especially when compared to state-of-the-art 2D detection systems [8, 46], which can detect hundreds of categories across various domains.

The performance gap between 2D and 3D detection, particularly in generalising to more detectable categories, is mostly due to i) the complex problem setup and ii) insufficient training data. The 3D detection task is closely related to various other tasks such as 2D detection, depth estimation and object pose estimation, each of which is a challenging research area. Additionally, labelling 3D data is more labour-intensive, requiring specifying nine degrees of freedom instead of four in 2D. This combination of complexity and limited data restricts current 3D detection methods to fewer categories, often with reduced accuracy.

To overcome these limitations, we propose a pipeline that decouples the 3D detection task from 2D detection and depth prediction. This decoupling enhances training efficiency, and, most importantly, allows for a category-agnostic approach with improved accuracy.

Our key idea lies in recovering a 3D bounding box from a random noise, conditioned on several visual prompts, using a denoising network inspired by diffusion models [20, 54] in a generative fashion. Specifically, the random noise is sampled from a normal distribution, with visual prompts consisting of the image of the target object, a 2D detection bounding box, and the depth of the target. During training, we take advantage of ground truth labels of 2D bounding boxes and object depths. During inference, the model can be integrated with 2D detectors and depth estimation models or take prompts from various sources, *e.g.* human annotation.

In addition to simplifying existing pipelines and achieving category-agnostic detection, our diffusion-based approach allows us to generate an *arbitrary* number of predictions for a *single* target due to their stochastic nature. We leverage this by estimating multiple 3D bounding boxes for one target, assigning a confidence score to each, and selecting the most confident one, further improving detection accuracy.

While developing our novel 3D detection method, we discovered that conventional metrics, such as Intersection over Union (IoU) and Generalised IoU (GIoU), often struggle to evaluate 3D detection results accurately, particularly for non-overlapping or enclosing cases, which are common for thin and small objects. To address these limitations, we propose a new metric called *Normalised Hungarian Distance* (NHD), which seeks a one-to-one assignment between the corners of the ground truth and predicted 3D bounding boxes, then calculates the Euclidean distance between corresponding corners, offering a more detailed and precise evaluation of 3D object detection.

In summary, we make three key contributions: First, we introduce a novel diffusion-based pipeline for 3D object detection that decouples the 3D detection task from 2D detection and depth prediction, enabling category-agnostic 3D detection. Second, we enhance the 3D detection accuracy by leveraging generative capabilities in our diffusion-based pipeline to predict multiple bounding boxes with confidence scores. Third, we propose the *Normalised Hungarian Distance* (NHD), a new evaluation metric that provides a more precise assessment of 3D detection results.

As a result, our method achieves state-of-the-art accuracy in 3D object detection in a category-agnostic manner and demonstrates strong generalisation to unseen datasets.

2. Related Work

2D Object Detection. 2D object detectors include two-stage detectors [15, 47] that use a coarse-to-fine approach and single-stage detectors [8, 34, 46, 47] that directly estimate the location of objects from the extracted visual features. DiffusionDet [9] was the first to apply diffusion to detection tasks, progressively refining noisy 2D boxes towards the target objects. Category-agnostic 2D detection models are conceptually similar to our approach in that they avoid using category information. These models learn to differentiate generic objects from the image background using low-level visual cues [41, 72] or using supervision from bounding box labels [25, 26, 38]. However, instead of learning objectness like these methods do, we focus on mapping 2D boxes to 3D boxes by leveraging visual cues from the input image.

Monocular 3D Object Detection Monocular 3D object detectors predict 3D cuboids from a single input image. Depending on the dataset domain, certain models are tailored for outdoor self-driving scenes [11, 23, 35, 37, 39, 39, 44, 45, 62], while others are specifically designed for indoor environments [24, 30, 40, 60]. Additionally, some studies [6, 32, 51] have explored integrating both indoor and outdoor datasets during training. These methods often use category labels for supervision [6, 23, 32, 35], require category information as a prior for initialisation or as input [6, 40, 44], or focus on specific scenes and object categories with strong assumptions about the predictions' dimensions or orientation [11, 39, 52, 64]. This reliance on category and scenario-specific knowledge limits their generalisation to in-the-wild scenes and novel categories. In contrast, our approach does not use category information during training or inference, focusing solely on predicting 3D bounding boxes. This allows the model to be used for novel objects that were not present during training.

Diffusion Models for Visual Perception Diffusion models [20, 53, 56, 57] have demonstrated remarkable results in computer vision [18, 21, 66], natural language processing [3, 17, 31, 67], and multimodal data generation [4, 42, 43, 50, 68]. In visual perception tasks, DiffusionDet [9] was the first to apply box diffusion for 2D object detection from a single RGB image. Additionally, diffusion has been utilised for tasks like image segmentation [2, 10] and human pose estimation [16, 22]. For 3D object detection, Zhou *et al.* [70] introduced diffusion of 5 DoF Bird's Eye View boxes as proposals for detection from point clouds. Diffusion-SS3D [19] employs diffusion for semi-supervised object detection in point clouds, denoising object size and class labels. DiffRef3D [27] and DiffuBox [12] apply diffusion to refine proposals/coarse boxes for 3D object detection from point clouds. MonoDiff [44] uses Gaussian Mixture Models to initialise the dimensions and poses of 3D bounding boxes, recovering the target boxes through diffusion conditioned on an image. Unlike these approaches, which assume an initial distribution or proposals for the diffusing box parameters [12, 44], limited DoFs for box orientation and dimensions [12, 27], or diffuse only partial parameters [19, 44], our model initialise all box parameters with random noise for diffusion. By conditioning the model on an image and a 2D box, our model recovers using diffusion the in-plane translations, three sizes, and three DoF for the rotation of the 3D box.

3. Method

Given an image I, a 2D bounding box of an object B, the object depth z, and the camera intrinsics K, our objective is to estimate the centre, 3D size and orientation of a 3D box that tightly encloses the object. We formulate this task as a conditional diffusion process [20], progressively recovering a target box from a noise sampled from normal distributions, conditioned on multiple prompts.

Let us consider a general diffusion setup first. In the forward diffusion process, Gaussian noise is incrementally added to a variable $\mathbf{x}_0$ over T time steps until it follows a normal distribution. During backward denoising, an estimate $\hat{\mathbf{x}}_0$ can be recovered from its noisy version $\mathbf{x}_t$ using a neural network f_θ:

$$f_\theta(\mathbf{x}_t, t) \rightarrow \hat{\mathbf{x}}_0, \tag{1}$$

where $t \in [1, T]$ denotes a diffusion step. *In our 3D object detection task, we consider* $\mathbf{x}_0$ *the parameters of a 3D box.*

To adapt the general diffusion process to this vision-based 3D detection setup, we consider a conditional denoising network f_θ:

$$f_\theta(\mathbf{x}_t, t, \mathbf{c}) \rightarrow \hat{\mathbf{x}}_0, \tag{2}$$

where $\mathbf{c}$ denotes the conditional signal that includes information from the input image I, the 2D bounding box B, the camera intrinsics K, and the object depth z. This conditional diffusion process is similar to diffusion-based text-

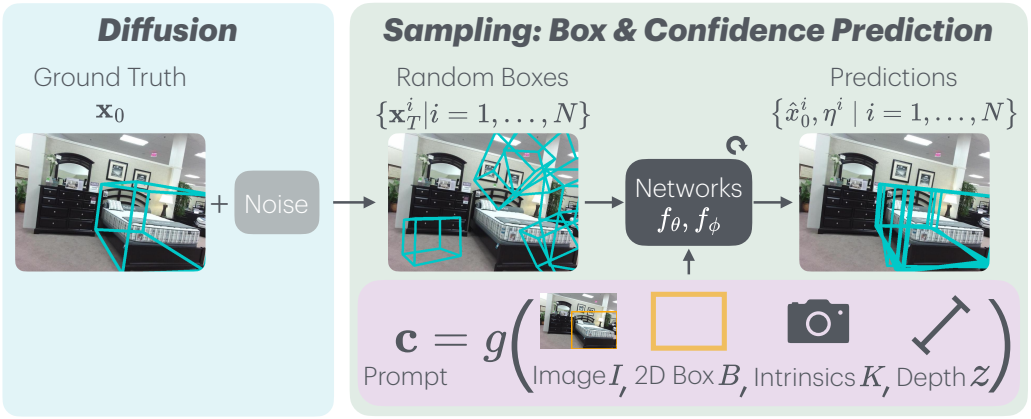


Figure 1. **Method Overview.** During forward diffusion, we add N independent Gaussian noises to a ground truth box $\mathbf{x}_0$ to obtain a number of noisy boxes. We then train a denoising network f_θ to recover the target box parameters $\hat{\mathbf{x}}_0$ from noisy boxes, conditioned on a vision-related signal $\mathbf{c}$. Additionally, we train another network f_ϕ to estimate a confidence score η for each predicted box. The final output is the box with the highest confidence score.

to-image generation tasks [50, 68], where the conditional signal typically consists of text descriptions.

In Sec. 3.1 we begin by introducing the parametrisation of the 3D bounding boxes and the prompt encoding. We then explain the forward diffusion and reverse sampling in Sec. 3.2 and Sec. 3.3 respectively, detailing how we utilise the generative properties of diffusion networks to predict multiple proposals and their associated confidence scores. Finally, we outline our training processes and losses in Sec. 3.4.

3.1. Preparation

Box Parameterisation We consider the position, orientation and size of a 3D bounding box. Specifically, each 3D bounding box $\mathbf{x}_0$ is represented using 11 parameters:

$$\mathbf{x} := [u, v, w, h, l, \mathbf{p}]. \tag{3}$$

The position of the 3D box is defined by the 2D projected coordinates u and v, which represent the object centre on the image plane of the observing camera. This position parametrisation decouples the image plane position components and the depth component, allowing us to leverage object depth from various sources. The orientation of the object relative to the camera of the input image is represented by the continuous 6D allocentric rotation $\mathbf{p} \in \mathbb{R}^6$ [71]. The size of the 3D bounding box is captured by w, h, and l. Overall, this parametrisation follows [6], but with the depth component excluded.

Prompts Encoding Our conditioning signal $\mathbf{c}$ is derived from the image I, a 2D bounding box B, camera intrinsics K, and the object depth z:

$$\mathbf{c} = g(I, B, K, z), \tag{4}$$

where function $g(\cdot)$ denotes a prompt encoding function. This function includes an image encoding backbone, positional encoding functions, and a shallow MLP that summaries all prompt information in preparation for the box prediction network. Further details on the prompt encoding can be found in the supplementary material.

3.2. Diffusion: Adding Noise to a Box

In the forward diffusion process, we start with a noise-free 3D bounding box, denoted as $\mathbf{x}_0$, and iteratively add Gaussian noise over T steps to generate a fully noisy box $\mathbf{x}_T$, which follows a normal distribution. This process follows the standard DDPM schedule [20].

Preprocessing Before applying noise to the original 3D bounding box $\mathbf{x}_0$, we perform additional normalisation and scaling to ensure that $\mathbf{x}_0$ lies within the range $[-s, s]$, where s corresponds to the signal-to-noise ratio of the diffusion process. During the normalisation step, the projected coordinates u and v are normalised relative to the image dimensions. The dimensions of the 3D bounding box—width w, height h, and length l—are normalised against a predefined maximum box size. The orientation of the box $\mathbf{p}$, expressed in the allocentric representation [71], is inherently normalised and does not require further adjustment. In the scaling step, all box parameters are further scaled by a scalar s. As shown in Sec. 5.5, this scaling step, adopted from [9, 10], improves box prediction accuracy.

3.3. Sampling: Predicting a Box

Our method predicts a 3D bounding box for a target object using a denoising network f_θ, conditioned on a vision-related prompt $\mathbf{c}$. To enhance detection performance, we introduce a confidence score for each predicted box, gen-

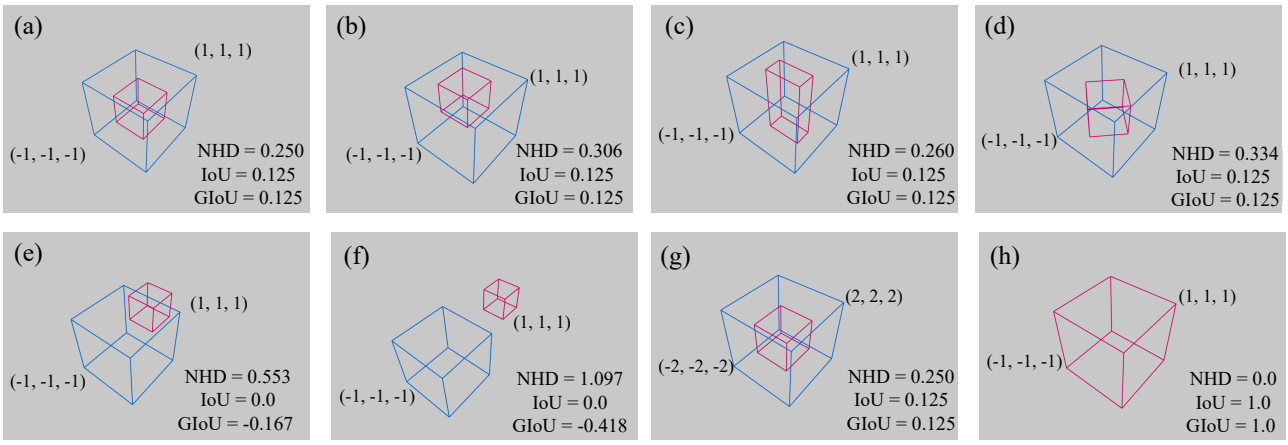


Figure 2. **Comparison Between NHD, IoU and GIoU**. Comparing block (a) with blocks (b, c, d, e, f), we show that NHD provides a more accurate measurement of errors compared to IoU and GIoU, particularly under translation, scaling, and rotation transformations. Block (g) demonstrates that all three metrics are scale-invariant. Block (h) presents metric values when the two boxes are perfectly aligned.

erating multiple candidate boxes and selecting the one with the highest confidence.

Single Box Prediction To predict a single 3D box $\hat{\mathbf{x}}_0$, we begin with a randomly sampled noise $\mathbf{x}_T$ and iteratively refine the 3D box $\hat{\mathbf{x}}_t$ with our denoising network f_θ, conditioned on the encoded prompt $\mathbf{c}$. This process continues until the final denoising step $t = 0$, following the standard sampling procedure introduced in DDIM [54].

Multi-Box Prediction and Selection Leveraging the generative capabilities of diffusion networks, our method allows for the prediction of *multiple* 3D boxes for a *single* target object. Specifically, for each target object, we sample N 3D box parameters $\{\mathbf{x}_T^i | i = 1, \ldots, N\}$ from a normal distribution, producing N predictions $\{\hat{\mathbf{x}}_0^i | i = 1, \ldots, N\}$. We introduce a learnable confidence score η^i for each prediction, where we select the box with the highest confidence score as the final prediction during inference.

Confidence Prediction To estimate the uncertainty $\mu \in (0, \infty)$ for each box prediction, we employ an additional network branch f_ϕ, which takes the current box estimation and the vision conditioning signal $\mathbf{c}$ as inputs:

$$f_\phi(\mathbf{c}, \mathbf{x}_t) \rightarrow \mu. \tag{5}$$

The confidence score $\eta \in (0, 1)$ is derived from the uncertainty through an exponential mapping $\eta = e^{-\mu}$. This score reflects the agreement between the current box estimation and the provided vision prompt.

3.4. Training

We follow the standard DDPM [20] training process and utilise a training loss $\mathcal{L}$ consisting of two loss terms:

$$\mathcal{L} = \mathcal{L}_{\text{3D}} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}}, \tag{6}$$

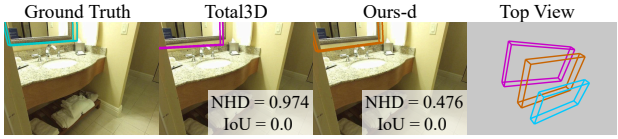


Figure 3. **IoU and NHD in a Practical Example.** For thin objects like mirrors, even a small translational offset can lead to an IoU of 0. In contrast, NHD effectively captures and reflects the box estimation error in these cases.

with λ_{reg} being a hyperparameter that balances the regularisation term $\mathcal{L}_{\text{reg}}$ and the reconstruction term $\mathcal{L}_{\text{3D}}$.

The reconstruction loss $\mathcal{L}_{\text{3D}}$ is designed to encourage our denoising network to predict accurate 3D boxes. This is achieved by penalising the Chamfer distance between the corners of predicted boxes and the ground truth, weighted by the confidence η:

$$\mathcal{L}_{\text{3D}} = \frac{1}{N} \sum_{i=1}^{N} \eta^i \mathcal{L}_{\text{chamfer}}(\hat{\mathbf{x}}_0^i, \mathbf{x}_0), \tag{7}$$

whereas the regularisation term $\mathcal{L}_{\text{reg}}$ prevents the uncertainty prediction μ being excessively large:

$$\mathcal{L}_{\text{reg}} = \frac{1}{N} \sum_{i=1}^{N} \mu^i; \tag{8}$$

Details regarding the Chamfer distance $\mathcal{L}_{\text{chamfer}}$ between two 3D boxes are provided in the supplementary material.

4. Metric: Normalised Hungarian Distance

Intersection-over-Union (IoU) is a common metric for evaluating 3D object detection [6, 40, 51]. While IoU is scale-invariant, it fails to measure the closeness of predictions

Table 1. **Detection Performance: Comparing Our Method with Cube R-CNN [6] and Total3D [40] on Omni3D Dataset.** The top three rows use GT 2D boxes along with predicted depths. The depths of our predictions are set to the same as [6] for fair comparison. The bottom three rows use GT 2D boxes and GT depths for all methods.

Methods	SUN RGB-D		Omni3D Indoor		Omni3D Indoor & Outdoor	
	IoU (%) ↑	NHD ↓	IoU (%) ↑	NHD ↓	IoU (%) ↑	NHD ↓
Total3D	24.8	0.376	-	-	-	-
Cube R-CNN	36.2	0.236	19.5	0.667	23.0	0.593
Ours-d	**40.2**	**0.231**	**20.8**	**0.648**	**23.3**	**0.591**
Total3D*	46.6	0.184	-	-	-	-
Cube R-CNN*	54.5	0.137	41.0	0.189	45.8	0.167
Ours*	**61.4**	**0.114**	**49.7**	**0.143**	**51.4**	**0.142**

Table 2. **Generalisation Performance: Novel Categories.** We train the model on 31 object categories from the SUN RGB-D training set and evaluate its IoU on 7 unseen categories. To show the importance of estimating in-plane offsets and box orientation, we use a Unprojection baseline that converts GT 2D boxes to 3D with GT depth and dimensions, setting 3D rotation to zero degrees.

Methods	Trained on	sofa	table	cabinet	toilet	bathtub	door	oven	avg.
Unprojection	N/A	28.2	27.1	28.6	25.6	23.9	23.6	37.2	27.7
Ours	SUN RGB-D	**56.4**	**56.8**	**53.0**	**61.3**	**46.7**	**27.7**	**62.1**	**52.0**

when two boxes do not overlap, which is problematic for thin and small objects like mirrors or televisions. An example is shown in Fig. 3. Later, Generalised IoU (GIoU) [48] has been proposed to address this, but it still does not fully capture alignment in terms of centre, scale, and orientation, as seen in Fig. 2.

We propose a new metric, *Normalised Hungarian Distance* (NHD), to provide a more precise evaluation for 3D object detection. NHD is calculated as

$$\text{NHD}(\mathcal{M}_{\text{pred}}, \mathcal{M}_{\text{gt}}) = \frac{1}{d_{\text{gt}}} \sum_{(i,j)\in P} \|a_i - b_j\|_2, \qquad (9)$$

where P represents the optimal 1-to-1 mapping between predicted box corners $\mathcal{M}_{\text{pred}}$ and ground truth box corners $\mathcal{M}_{\text{gt}}$. The mapping P is obtained through a linear assignment algorithm by minimising the Euclidean distance between corresponding corners, ensuring corner $a_i \in \mathcal{M}_{\text{pred}}$ in the prediction matches corner $b_j \in \mathcal{M}_{\text{gt}}$ in the ground truth. To make NHD scale-invariant, we normalise it with the maximum diagonal length d_{gt} of the ground truth box.

5. Experiments

We outline our experimental setup in Sec. 5.1. In Secs. 5.2 and 5.3, we compare our approach to baselines, demonstrating its accuracy and generalisation. We also present its application in 3D dataset labelling (Sec. 5.4) and discuss the impact of hyperparameter choices (Sec. 5.5).

5.1. Experimental Setup

Datasets For training and evaluation, we utilise the Omni3D dataset [6], which is composed of six datasets[55], ARKitScenes [5], Hypersim [49], Objectron [1], KITTI [14], and nuScenes [7], encompassing both indoor and outdoor environments. For the quantitative tests in Sec. 5.4, we employ the COCO [33] and nerfstudio [59] datasets.

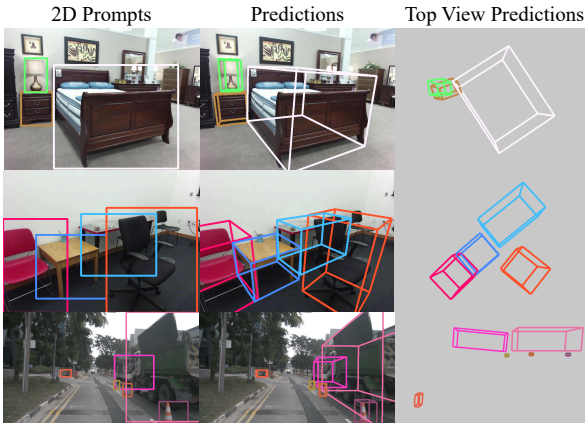


Figure 4. **Detection Performance: Results on Omni3D Test Set**. Estimating 3D box from GT 2D boxes and GT object depths.

Baselines To accurately assess the performance of our 3D detector, it is essential to eliminate errors from sources such as 2D detection and depth estimation. In Sec. 5.2, we assume ground truth 2D boxes and depth are provided for all methods during evaluation. We select Cube R-CNN [6] and Total3DUnderstanding [40] as our primary baselines, as their 3D detection heads can function independently. Additionally, since the baseline methods require category information, we provide ground truth category labels to their



Figure 5. **Generalisation Performance: Results for In-the-Wild Objects on COCO Dataset.** We show predictions made by our method without knowing object depths or camera intrinsics. By using constant values for depths and camera intrinsics, our approach accurately predicts 3D boxes with well-aligned projections on the image.

models.

Metrics In Sec. 5.2, we use IoU and NHD as two metrics to evaluate the performance of 3D detectors.

Model Our model architecture is built on Detectron2 [63]. We use the Swin Transformer [36] pre-trained on ImageNet-22K [13] as the image feature encoder and freeze all parameters during training. The denoising decoder adopts the iterative architecture in Sparse R-CNN [58] with 6 blocks, where the predictions from the previous stage are used as the input for the next stage. Predictions from each stage are used to compute loss against the ground truth.

Training & Inference We train the model for 270k iterations with a batch size of 16 on 2 A5000 GPUs. In comparison, Cube R-CNN is trained on 48 V100 GPUs with a batch size of 192. We use AdamW optimiser [28] with a learning rate of 2.5×10^{-5} which decays by a factor of 10 at 150k and 200k iterations. During training, we use image augmentation including random horizontal flipping and resizing. During evaluation, as the box parameters are initialised with random noise, the result can vary with different random seeds. To obtain a stable result, experiment results reported in Sec. 5.2 and Sec. 5.3 are computed by averaging the results of 10 different random seeds. We also analyse the stability of model performance with random seeds in the supplementary material.

5.2. Detection Performance

We compare our model with other 3D object detection approaches on Omni3D dataset [6] and its subsets. For baseline models [6, 40], we assume ground truth 2D boxes as the oracle 2D detection results and provide category information. In contrast, our method requires 2D boxes only and does not use the category information. As our model requires object depth as input, we use the depths estimated by [6] for a fair comparison. We introduce a variant of each method that substitutes estimated object depths with ground truth depths. As shown in Tab. 1, our approach outperforms the baselines, even with their use of additional category information and dimension priors. The comparison between using predicted depths (upper half of table) and ground truth depths (lower half) highlights that inaccurate monocular depth estimation is a key source of error in monocular 3D detection models. Fig. 4 shows the qualitative performance of our model on the Omni3D test set.

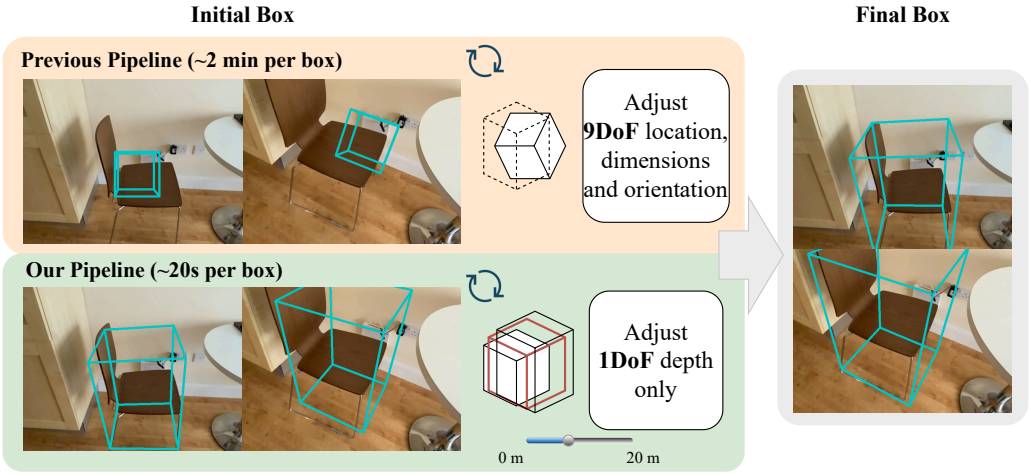


Figure 6. **Application: 3D Detection Dataset Annotation.** In a conventional 3D box annotation pipeline, annotators typically need to adjust a randomly initialised 3D box across nine degrees of freedom (rotation, translation, and size) until it appears correct in every view. This process is time-consuming and labour-intensive. Our model streamlines this workflow by reducing the task to a single degree of freedom, depth, significantly accelerating the dataset labelling process.

5.3. Generalisation Performance

Cross-category Generalisation To verify our model's generalisation ability across different objects, we trained it on the SUN RGB-D dataset using 31 out of 38 categories and evaluated it on the remaining categories. Since there are no previous baselines for comparison, we created a simple baseline using axis-aligned 3D boxes to demonstrate the importance of estimating in-plane offsets and box orientation, as well as the effectiveness of our approach. This baseline method takes the 2D box prompt and back-projects it to 3D using the provided depth and focal length. The bounding box orientation is set to match the camera orientation, and the dimension in the z-direction is set to the maximum length of the ground truth bounding box in that direction. Results in Tab. 2 and Fig. 7 show that our approach significantly outperforms this baseline and demonstrates good generalisation ability.

Cross-dataset Generalisation To assess the model's generalisation across different datasets and input types, we perform a qualitative analysis with Omni3D-trained models on test images from COCO and nerfstudio datasets.

Figure 5 shows the predictions on COCO, using only 2D prompts and uniform depth and camera intrinsics for all objects. Despite inaccuracies in depth and intrinsics, our model accurately estimates up-to-scale 3D boxes where the projections align with the objects in the image.

Figure 8 demonstrates the use of a monocular depth estimation network [65] and a segmentation network [29], combined with 2D prompts, to estimate 3D boxes with accurate top-view projections. The depth for each object is calculated by averaging the depths within the object's mask.

When 3D data, such as point clouds or CAD models, is available, it can also be utilised for depth inference. Figure 9 illustrates an example from nerfstudio dataset where 3D point clouds and a manually annotated 2D prompt are used to estimate the 3D box.

5.4. Application

As discussed in Sec. 1, the availability of 3D detection datasets is significantly more limited compared to 2D detection, primarily due to the high costs associated with annotating 3D bounding boxes. Figure 6 illustrates a typical 3D box annotation process, where annotators must draw a 3D box and fine-tune its projections across multiple views. This manual adjustment of the box's location, dimensions, and orientation can take several minutes per box.

Our model streamlines this process by reducing the complexity from adjusting nine degrees of freedom (DoF) to just one DoF – depth. Starting with a 2D prompt from an annotator, our model generates a 3D box prediction with depth ambiguity. The human annotator only needs to adjust the predicted box depth and verify it across other views. Incorporating this approach into the 3D box annotation pipeline has the potential to greatly improve annotation efficiency.

5.5. Discussion

We conducted a model analysis on the 10 common categories in the SUN RGB-D test set to study its performance. Unless otherwise stated, all inferences are run with 10 predictions for each object at a single sampling step.

Signal-to-noise Ratio In Tab. 3, we show the influence of the diffusion process's signal-to-noise ratio (SNR). Setting SNR to 2 achieves the highest performance in IoU and

Table 3. **Discussion: Model Analysis on SUN RGB-D Test Set.**

(a) IoU and NHD under various SNR (s).

SNR	IoU (%) ↑	NHD ↓
1.0	57.2	0.195
2.0	**59.8**	**0.177**
3.0	59.7	0.178

(b) IoU and NHD under various DDIM steps.

Iter steps	IoU (%) ↑	NHD ↓
1	61.19	0.1153
3	61.23	0.1150
5	**61.28**	**0.1147**

(c) IoU under various No. of sampled boxes.

$N_{\text{train}} \backslash N_{\text{eval}}$	1	10	100
1	58.1	60.2	60.2
10	60.9	61.2	61.2
100	61.2	**61.4**	**61.4**

NHD, which is consistent with the observations in other diffusion models for detection [9, 27].

Inference Iteration Steps During inference, we observed that increasing the number of DDIM sampling steps enhances model performance, albeit at the cost of longer inference times. As shown in Tab. 3b, raising the iteration steps from 1 to 5 results in performance gains, which plateau with additional steps.

Number of Sampled Boxes. Since we use random noise to initialise box parameters, the number of boxes sampled is flexible and can vary between training (N_{train}) and inference (N_{eval}). Table 3c presents the results for different combinations of N_{train} and N_{eval}. We observed that increasing N_{train} enhances model performance, with a cost of larger memory consumption. However, increasing N_{eval} beyond 10 does not yield additional improvement. Consequently, we select $N_{\text{train}} = 100$ and $N_{\text{eval}} = 10$ for other experiments.

6. Conclusion

Our diffusion-based pipeline significantly improves 3D object detection by decoupling it from 2D detection and depth prediction, enabling category-agnostic detection. Furthermore, the introduction of the Normalised Hungarian Distance (NHD) metric addresses the limitations of existing evaluation methods, providing a more accurate assessment of 3D detection outcomes, especially for complex scenarios involving small or thin objects. Experimental results con-

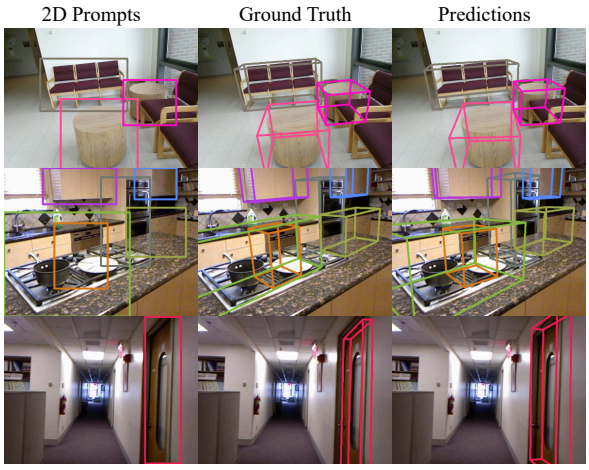


Figure 7. **Generalisation Performance: Predictions on Novel Categories**. We show that our model generalises well to unseen object categories.

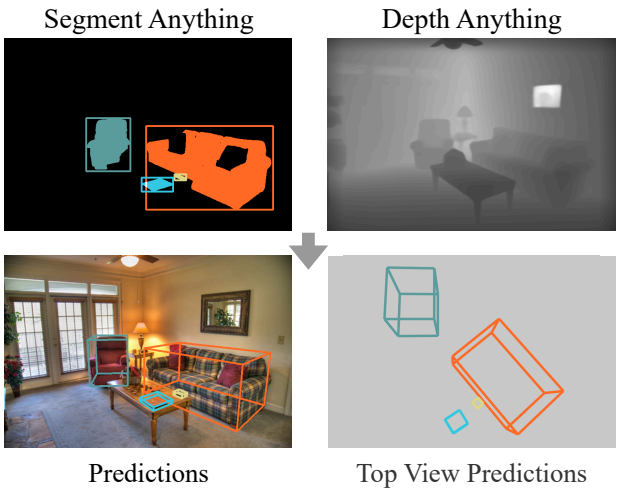


Figure 8. **Generalisation Performance: Predict with Prompts from 2D Detectors and Monocular Depth Estimators.** We infer object depths using DepthAnything [65] and SegmentAnything [29] for COCO Dataset.

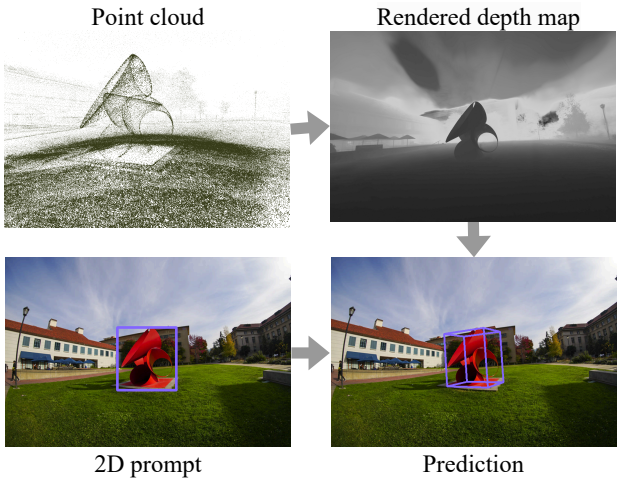


Figure 9. **Generalisation Performance: Predict with Prompts from human annotated 2D bounding boxes and a point cloud.**

firm that our method achieves state-of-the-art accuracy and strong generalisation across various object categories and datasets.

References

[1] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *CVPR*, 2021. 5

[2] Tomer Amit, Tal Shaharbany, Eliya Nachmani, and Lior Wolf. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*, 2021. 2

[3] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *NeurIPS*, 2021. 2

[4] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *CVPR*, 2022. 2

[5] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Yuri Feigin, Peter Fu, Thomas Gebauer, Daniel Kurz, Tal Dimry, Brandon Joffe, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. In *NeurIPS*, 2021. 5, 12

[6] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3D: A large benchmark and model for 3D object detection in the wild. In *CVPR*, 2023. 1, 2, 3, 4, 5, 6, 12, 14

[7] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 5

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 2

[9] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *ICCV*, 2023. 2, 3, 8

[10] Ting Chen, Lala Li, Saurabh Saxena, Geoffrey Hinton, and David J Fleet. A generalist framework for panoptic segmentation of images and videos. In *ICCV*, 2023. 2, 3

[11] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *CVPR*, 2016. 1, 2

[12] Xiangyu Chen, Zhenzhen Liu, Katie Z Luo, Siddhartha Datta, Adhitya Polavaram, Yan Wang, Yurong You, Boyi Li, Marco Pavone, Wei-Lun Chao, et al. Diffubox: Refining 3d object detection with point diffusion. *arXiv preprint arXiv:2405.16034*, 2024. 2

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 5

[15] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 2

[16] Jia Gong, Lin Geng Foo, Zhipeng Fan, Qiuhong Ke, Hossein Rahmani, and Jun Liu. Diffpose: Toward more reliable 3d pose estimation. In *CVPR*, 2023. 2

[17] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. In *ICLR*, 2023. 2

[18] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. *NeurIPS*, 2022. 2

[19] Cheng-Ju Ho, Chen-Hsuan Tai, Yen-Yu Lin, Ming-Hsuan Yang, and Yi-Hsuan Tsai. Diffusion-ss3d: Diffusion model for semi-supervised 3d object detection. *NeurIPS*, 2024. 2

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 1, 2, 3, 4

[21] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *NeurIPS*, 2022. 2

[22] Karl Holmquist and Bastian Wandt. Diffpose: Multi-hypothesis human pose estimation using diffusion models. In *CVPR*, 2023. 2

[23] Kuan-Chih Huang, Tsung-Han Wu, Hung-Ting Su, and Winston H Hsu. Monodtr: Monocular 3d object detection with depth-aware transformer. In *CVPR*, 2022. 2

[24] Siyuan Huang, Siyuan Qi, Yinxue Xiao, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu. Cooperative holistic scene understanding: Unifying 3d object, layout, and camera pose estimation. *NeurIPs*, 2018. 2

[25] Ayush Jaiswal, Yue Wu, Pradeep Natarajan, and Premkumar Natarajan. Class-agnostic object detection. In *WACV*, 2021. 2

[26] Dahun Kim, Tsung-Yi Lin, Anelia Angelova, In So Kweon, and Weicheng Kuo. Learning open-world object proposals without learning to classify. *IEEE Robotics and Automation Letters*, 2022. 2

[27] Se-Ho Kim, Inyong Koo, Inyoung Lee, Byeongjun Park, and Changick Kim. Diffref3d: A diffusion-based proposal refinement framework for 3d object detection. *arXiv preprint arXiv:2310.16349*, 2023. 2, 8

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[29] 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 *ICCV*, 2023. 7, 8

[30] Nilesh Kulkarni, Ishan Misra, Shubham Tulsiani, and Abhinav Gupta. 3d-relnet: Joint object and relational network for 3d prediction. In *ICCV*, 2019. 2

[31] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *NeurIPS*, 2022. 2

[32] Zhuoling Li, Xiaogang Xu, SerNam Lim, and Hengshuang Zhao. Unimode: Unified monocular 3d object detection. In *CVPR*, 2024. 1, 2

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 5

[34] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 2

[35] Yuxuan Liu, Yuan Yixuan, and Ming Liu. Ground-aware monocular 3d object detection for autonomous driving. *IEEE Robotics and Automation Letters*, 2021. 2

[36] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 6, 12

[37] Xinzhu Ma, Zhihui Wang, Haojie Li, Pengbo Zhang, Wanli Ouyang, and Xin Fan. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In *ICCV*, 2019. 2

[38] Muhammad Maaz, Hanoona Rasheed, Salman Khan, Fahad Shahbaz Khan, Rao Muhammad Anwer, and Ming-Hsuan Yang. Class-agnostic object detection with multi-modal transformer. In *ECCV*, 2022. 2

[39] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *CVPR*, 2017. 1, 2

[40] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *CVPR*, 2020. 1, 2, 4, 5, 6, 12

[41] Pedro O O Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. *NeurIPS*, 2015. 2

[42] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2022. 2

[43] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[44] Yasiru Ranasinghe, Deepti Hegde, and Vishal M Patel. Monodiff: Monocular 3d object detection and pose estimation with diffusion models. In *CVPR*, 2024. 2

[45] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *CVPR*, 2021. 2

[46] Dillon Reis, Jordan Kupec, Jacqueline Hong, and Ahmad Daoudi. Real-time flying object detection with yolov8. *arXiv preprint arXiv:2305.09972*, 2023. 1, 2

[47] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 2015. 2

[48] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019. 5

[49] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, 2021. 5, 12

[50] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. 2022 ieee. In *CVPR*, 2021. 2, 3

[51] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In *WACV*, 2022. 1, 2, 4

[52] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *ICCV*, 2019. 1, 2

[53] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 2

[54] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2020. 1, 4

[55] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015. 5

[56] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 2019. 2

[57] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2020. 2

[58] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *CVPR*, 2021. 6

[59] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *SIGGRAPH 2023*, 2023. 5

[60] Shubham Tulsiani, Saurabh Gupta, David F Fouhey, Alexei A Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. In *CVPR*, 2018. 2

[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 12

[62] Tai Wang, ZHU Xinge, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. In *CoRL*, 2022. 2

[63] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 6

[64] Longfei Yan, Pei Yan, Shengzhou Xiong, Xuanyu Xiang, and Yihua Tan. Monocd: Monocular 3d object detection with complementary depths. In *CVPR*, 2024. 2

[65] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024. 7, 8

[66] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *Entropy*, 2023. 2

[67] P Yu, S Xie, X Ma, B Jia, B Pang, R Gao, Y Zhu, S-C Zhu, and YN Wu. Latent diffusion energy-based model for interpretable text modeling. In *ICML 2022*, 2022. 2

[68] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 2, 3

[69] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3d object detection. In *CVPR*, 2021. 1

[70] Xin Zhou, Jinghua Hou, Tingting Yao, Dingkang Liang, Zhe Liu, Zhikang Zou, Xiaoqing Ye, Jianwei Cheng, and Xiang Bai. Diffusion-based 3d object detection with random boxes. In *PRCV*, 2023. 2

[71] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019. 3

[72] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 2

A. Implementation Details

A.1. Prompt Encoding

We encode the image I, a 2D bounding box B, camera intrinsics K, and the object depth z into the conditioning signal $\mathbf{c}$ through

$$\mathbf{c} = g(I, B, K, z), \tag{10}$$

The 2D bounding box B is described by the centre of the box along with its height and width on the image plane, i.e.

$$B := [u_{2d}, v_{2d}, w_{2d}, h_{2d}] \tag{11}$$

To account for variation in depth and focal length, we further unproject the width and height of the 2D box into 3D using the following equation:

$$(w_{3d}, h_{3d}) = (w_{2d}\frac{z}{f_x}, h_{2d}\frac{z}{f_y}), \tag{12}$$

where f_x and f_y are the focal lengths from the intrinsics K.

For the input image I, we first encode it with a pre-trained Swin Transformer [36] to generate multi-scale feature maps F. Next, we extract local image features inside the region of the 2D box prompt to obtain F_{RoI}. Additionally, we apply a cross-attention layer [61] between F and the 2D box B to obtain F_{atten}.

By concatenating the transformed box prompt, image features and the object depth, the final conditioning signal $\mathbf{c}$ can be written as

$$\mathbf{c} = [F_{\text{RoI}}, F_{\text{attn}}, u_{2d}, v_{2d}, w_{3d}, h_{3d}, z]. \tag{13}$$

A.2. Loss Function

The Chamfer distance between the corners of the predicted 3D boxes $\mathcal{M}_{\text{pred}} = \{a_i | i = 1...8\}$ and the corners of the ground truth boxes $\mathcal{M}_{\text{gt}} = \{b_i | i = 1...8\}$ is computed as

$$\mathcal{L}_{\text{chamfer}} = \sum_{a_i \in \mathcal{M}_{\text{pred}}} \min_{b_i \in \mathcal{M}_{\text{gt}}} \|a_i - b_i\|_1 + \sum_{b_i \in \mathcal{M}_{\text{gt}}} \min_{a_i \in \mathcal{M}_{\text{pred}}} \|b_i - a_i\|_1. \tag{14}$$

A.3. Baseline Models

Unprojection Fig. 12 illustrates how we obtain the Unprojection baseline for experiments in Sec.5.3 of the main paper.

Total3DUnderstanding [40] We use their publicly released code and the model pre-trained on SUN RGB-D in experiments of Section 5.

Cube R-CNN [6] For the results on the SUN RGB-D dataset in the second row of Table 1 in the main paper, we use the numbers reported directly from their paper. For the other experiments in Section 5, we use their publicly available code and pre-trained models.

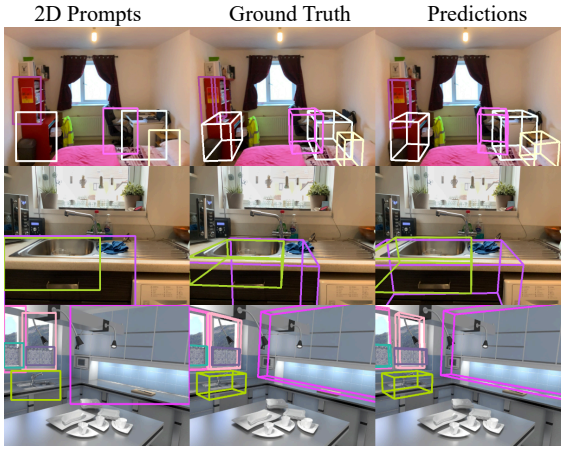


Figure 10. **Generalisation Performance: Results for Cross-Dataset Test.** We show predictions on ARKitScenes and Hypersim made by our method trained on SUN RGB-D.

Table 4. **Randomness Analysis on SUN RGB-D test set.** We evaluate the model using 10 different random seeds and report the mean, maximum, minimum, and standard deviation σ for both IoU and NHD.

	Mean	Max	Min	σ
IoU (%) ↑	61.38	61.46	61.24	6.4e-4
NHD ↓	0.1140	0.1146	0.1133	3.9e-4

A.4. Algorithms

The training and inference algorithms are shown in Algorithm 1 and Algorithm 2.

B. Additional Results

B.1. Generalisation

Cross-dataset generalisation Our model trained on SUN RGB-D achieves an average IoU of 39.0 on the Hypersim [49] test set and 48.2 on the ARKitScenes [5] test set, highlighting its strong generalisation across different datasets. Fig. 10 presents some of the test results.

Additional COCO results Fig. 11 shows additional results on COCO dataset.

B.2. Per-category SUN RGB-D Performance

In Table 1 of the main paper, we report the average IoU and NHD on 10 common categories of the SUN RGB-D dataset to make a fair comparison for baseline methods with different categories. Tab. 5 and Tab. 6 show the per-category IoU and NHD performances respectively.

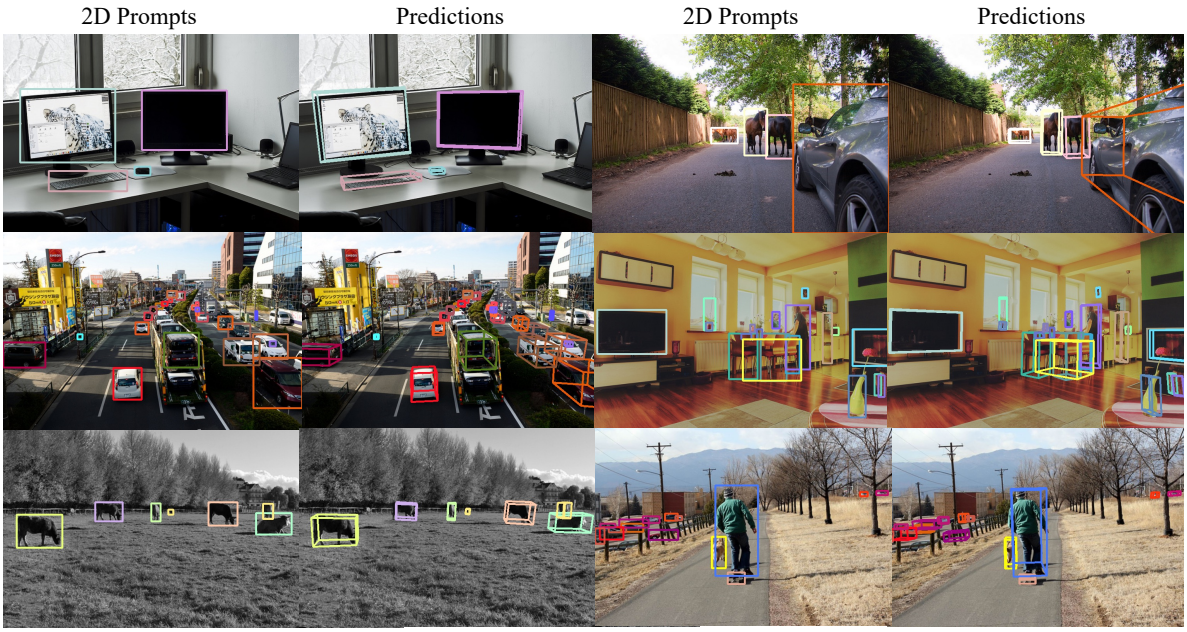


Figure 11. **Generalisation Performance: Additional Results for In-the-Wild Objects on COCO Dataset.** We show predictions made by our method without knowing object depths or camera intrinsics. By using constant values for depths and camera intrinsics, our approach accurately predicts 3D boxes with well-aligned projections on the image.



Figure 12. **Unprojection Baseline Illustration.** The Unprojection baseline (green) converts GT 2D boxes to 3D using GT depth and dimensions that match the GT 3D box (blue), with the 3D rotation to zero degrees.

B.3. Randomness Analysis

As discussed in Section 5.1 of the main paper, the diffusion process involves inherent randomness, so we conducted the experiments using 10 different random seeds and report the averaged results. To assess the model's stability, in addition to the averaged value reported in the main paper, we also provide the maximum, minimum, and standard deviation across these 10 runs in Tab. 4.

B.4. Noise on 2D Box

We analyse the model's robustness towards noise during inference in Tab. 7. We simulate box noise by applying Gaussian noise to box scales and translations separately, which can be written as:

$$w' = w + \mathcal{N}(0, \sigma^2_{\text{scale}}) \quad h' = h + \mathcal{N}(0, \sigma^2_{\text{scale}}), \tag{15}$$

$$x' = x + \mathcal{N}(0, \sigma^2_{\text{trans}} \cdot w) \quad y' = y + \mathcal{N}(0, \sigma^2_{\text{trans}} \cdot h), \tag{16}$$

where w, h are the height and width of the ground truth boxes, x, y are the centre coordinates and w', h', x', y' are the noisy parameters. σ^2_{scale} and σ^2_{trans} are the variances of scale and translation noise. Tab. 7 shows that while the model is robust to noise in box scale and translation, translation errors have a greater impact on accuracy.

Table 5. **Per-category IoU (%) on SUN RGB-D test set.** The top three rows use GT 2D boxes along with predicted depths. The depths of our predictions are set to the same as [6] for fair comparison. The bottom three rows use GT 2D boxes and GT depths for all methods.

Methods	Trained on	table	bed	sofa	bathtub	sink	shelves	cabinet	fridge	chair	toilet	avg.
Total3D	SUN RGB-D	28.0	37.0	30.1	27.6	20.1	10.8	14.3	20.2	24.8	35.4	24.8
Cube R-CNN	SUN RGB-D	39.2	49.5	46.0	32.2	31.9	16.2	26.5	34.7	39.9	45.7	36.2
Cube R-CNN	Omni3D Indoor	41.4	50.9	**50.8**	**39.2**	35.0	17.8	28.2	**35.1**	41.3	48.1	38.8
Ours-d	SUN RGB-D	**42.2**	**54.4**	50.5	38.9	**40.3**	**19.7**	**29.4**	33.5	**43.2**	**50.1**	**40.2**
Total3D*	SUN RGB-D	45.0	47.9	49.7	49.5	44.8	30.8	38.2	48.2	56.3	55.8	46.6
Cube R-CNN*	Omni3D Indoor	54.8	57.0	62.9	52.7	49.7	37.5	47.6	58.5	63.6	61.7	54.5
Ours*	SUN RGB-D	**63.1**	**64.3**	**64.8**	**56.7**	**62.6**	**44.0**	**56.5**	**62.2**	**70.3**	**68.9**	**61.4**

Table 6. **Per-category NHD on SUN RGB-D test set.** The top three rows use GT 2D boxes along with predicted depths. The depths of our predictions are set to the same as [6] for fair comparison. The bottom three rows use GT 2D boxes and GT depths for all methods.

Methods	Trained on	table	bed	sofa	bathtub	sink	shelves	cabinet	fridge	chair	toilet	avg.
Total3D	SUN RGB-D	0.352	0.254	0.314	0.288	0.526	0.497	0.443	0.380	0.408	0.297	0.376
Cube R-CNN	Omni3D Indoor	0.230	0.162	**0.164**	**0.215**	0.244	0.324	0.384	**0.229**	0.233	0.172	0.236
Ours-d	SUN RGB-D	**0.219**	**0.156**	0.167	0.219	**0.231**	**0.322**	**0.372**	0.233	**0.230**	**0.162**	**0.231**
Total3D*	SUN RGB-D	0.204	0.168	0.180	0.157	0.188	0.251	0.210	0.177	0.148	0.160	0.184
Cube R-CNN*	Omni3D Indoor	0.148	0.125	0.107	0.149	0.146	0.181	0.176	0.117	0.107	0.112	0.137
Ours*	SUN RGB-D	**0.114**	**0.107**	**0.101**	**0.120**	**0.114**	**0.161**	**0.127**	**0.111**	**0.093**	**0.090**	**0.114**

Table 7. **Noise on the 2D box**. We add different levels of random noise to the scale and translation of the 2D object box and report the model performance with these noisy box inputs.

σ_{scale}	σ_{trans}	IoU (%) ↑	NHD ↓
0.00	0.00	61.4	0.114
0.05	0.00	59.9	0.120
0.00	0.05	56.1	0.132
0.05	0.05	55.2	0.135
0.10	0.10	46.1	0.174

Algorithm 1: Training

```
def train_loss(images, gt_cubes, boxes_2d):
    """
    images: [B, H, W, 3]
    gt_cubes: [B, 1, D]
    boxes_2d: [B, 4]

    D: dimension of cubes
    N_train: number of sampled boxes during
        training
    """

    # Encode image features
    feats = image_encoder(images)

    # Separate depth information
    # from cube parameters
    cube_params, depths = separate_depth(gt_cubes
        )

    # normalise cube_params to [0, 1]
    cube_params = normalise_cube(cube_params)

    # Duplicate cube_params to N_train
    x_0 = duplicate_cubes(cube_params)

    # Signal scaling
    x_0 = (x_0 * 2 - 1) * scale

    # Corrupt x_0
    t = randint(0, T) # time step
    eps = normal(mean=0, std=1) # noise: [B,
        N_train, D-1]
    x_t = (
        sqrt(alpha_cumprod(t)) * x_0
        + sqrt(1 - alpha_cumprod(t)) * eps
    )

    # Predict
    x_0_pred = denoising_model(
        x_t, feats, t, boxes_2d, depths
    )

    # Set prediction loss
    loss = L(x_0_pred, gt_cubes)

    return loss
```

Algorithm 2: Inference

```
def infer(images, steps, T, boxes_2d, depths):
    """
    images: [B, H, W, 3]
    steps: number of sampling steps
    T: total number of time steps
    boxes_2d: [B, 4]
    depths: object depths [B, 1]

    N_eval: number of proposal boxes during
        inference
    """

    # Encode image features
    feats = image_encoder(images)

    # Initialise noisy cube parameters (excluding
        depth) [B, N_eval, D-1]
    x_t = normal(mean=0, std=1)

    # Define uniform sampling step sizes
    times = reversed(
        linspace(0, T, steps)
    )

    # Generate pairs of consecutive time steps
    time_pairs = list(
        zip(times[:-1], times[1:])
    )

    # Iterate through time pairs
    for t_now, t_next in time_pairs:
        # Predict cube parameters x_0 from x_t
        x_0_pred = denoising_model(
            x_t, feats, t_now, boxes_2d, depths
        )

        # Estimate x_t at t_next
        x_t = ddim_step(
            x_t, x_0_pred, t_now, t_next
        )

    # Combine predicted cube parameters with
        depth information
    pred_cubes = combine_cubes(x_0_pred, depths)

    return pred_cubes
```

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	CatFree3D: Category-agnostic 3D Object Detection with Diffusion
Publication Status	☑Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	**Bian, W**., Wang, Z., & Vedaldi, A. (2024). CatFree3D: Category-agnostic 3D Object Detection with Diffusion. In *International Conference on 3D Vision (3DV).*

Student Confirmation

Student Name:	Wenjing Bian		
Contribution to the Paper	• Conception and validation of the idea • Design of the architecture and framework • Implementation and running of all experiments • Research of prior work • Writing and presentation of the paper		
Signature	Wenjing Bian	Date	April 22, 2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Victor Adrian Prisacariu		
Supervisor comments The description is accurate		
Signature VPA	Date	April 22, 2025

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 5

NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior

As a per-scene optimisation approach, Neural Radiance Fields (NeRF) demonstrate impressive performance in synthesising photorealistic images. However, NeRF suffers from scale–radiance ambiguity [79] due to the lack of geometric information and reliance solely on 2D photometric losses. When camera poses are unknown, this ambiguity becomes even more pronounced. Existing approaches such as NeRFmm [36] and BARF [37] attempt to jointly optimise camera poses and scene geometry but often struggle with complex camera trajectories, tending to overfit to input images and failing to recover correct geometry and poses.

In this chapter, we introduce NoPe-NeRF, an approach for optimising NeRF without any prior knowledge of camera poses. To alleviate the inherent ambiguities, we introduce geometric prior from off-the-shelf monocular depth maps as additional constraints during optimisation. These depth maps serve two roles: first, they regularise the depth predicted by the NeRF model; second, they constrain

relative camera poses through inter-frame geometric and photometric consistency losses. Recognising that monocular depth maps can exhibit distortions due to scale ambiguity, we further parameterise per-frame depth distortions and jointly optimise them alongside the NeRF model and camera poses. Our method outperforms prior approaches in terms of pose accuracy, novel view synthesis quality, and depth prediction accuracy across a range of indoor and outdoor datasets.

To summarise, our main contributions are:

- We propose a novel approach to incorporate depth priors, with distortion modelling, into the optimisation of NeRF with unknown camera poses.
- We design inter-frame geometric and photometric loss functions to regularise relative pose estimation.
- We achieve superior performance compared to previous approaches in view synthesis quality, pose accuracy, and depth prediction across multiple challenging datasets.

The rest of this chapter is based on our paper *NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior* presented at *Computer Vision and Pattern Recognition (CVPR) 2023*.

NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior

Wenjing Bian Zirui Wang Kejie Li Jia-Wang Bian
Victor Adrian Prisacariu
Active Vision Lab, University of Oxford
{wenjing, ryan, kejie, jiawang, victor}@robots.ox.ac.uk



Figure 1. **Novel view synthesis comparison**. We propose NoPe-NeRF for joint pose estimation and novel view synthesis. Our method enables more robust pose estimation and renders better novel view synthesis than previous state-of-the-art methods.

Abstract

Training a Neural Radiance Field (NeRF) without pre-computed camera poses is challenging. Recent advances in this direction demonstrate the possibility of jointly optimising a NeRF and camera poses in forward-facing scenes. However, these methods still face difficulties during dramatic camera movement. We tackle this challenging problem by incorporating undistorted monocular depth priors. These priors are generated by correcting scale and shift parameters during training, with which we are then able to constrain the relative poses between consecutive frames. This constraint is achieved using our proposed novel loss functions. Experiments on real-world indoor and outdoor scenes show that our method can handle challenging camera trajectories and outperforms existing methods in terms of novel view rendering quality and pose estimation accuracy. Our project page is https://nope-nerf.active.vision.

1. Introduction

The photo-realistic reconstruction of a scene from a stream of RGB images requires both accurate 3D geometry reconstruction and view-dependent appearance modelling. Recently, Neural Radiance Fields (NeRF) [24] have demonstrated the ability to build high-quality results for generating photo-realistic images from novel viewpoints given a sparse set of images.

An important preparation step for NeRF training is the estimation of camera parameters for the input images. A current go-to option is the popular Structure-from-Motion (SfM) library COLMAP [35]. Whilst easy to use, this pre-processing step could be an obstacle to NeRF research and real-world deployments in the long term due to its long processing time and its lack of differentiability. Recent works such as NeRFmm [47], BARF [18] and SC-NeRF [12] propose to simultaneously optimise camera poses and the neural implicit representation to address these issues. Nevertheless, these methods can only handle forward-facing scenes when no initial parameters are supplied, and fail in dramatic camera motions, *e.g.*a casual handheld captured video.

This limitation has two key causes. First, all these methods estimate a camera pose for each input image individually without considering relative poses between images. Looking back to the literature of Simultaneous localisation and mapping (SLAM) and visual odometry, pose estimation can significantly benefit from estimating relative poses be-

tween adjacent input frames. Second, the radiance field is known to suffer from *shape-radiance* ambiguity [56]. Estimating camera parameters jointly with NeRF adds another degree of ambiguity, resulting in slow convergence and unstable optimisation.

To handle the limitation of large camera motion, we seek help from monocular depth estimation [22, 28, 29, 52]. Our motivation is threefold: First, monocular depth provides strong geometry cues that are beneficial to constraint *shape-radiance* ambiguity. Second, relative poses between adjacent depth maps can be easily injected into the training pipeline via Chamfer Distance. Third, monocular depth is lightweight to run and does not require camera parameters as input, in contrast to multi-view stereo depth estimation. For simplicity, we use the term *mono-depth* from now on.

Utilising mono-depth effectively is not straightforward with the presence of scale and shift distortions. In other words, mono-depth maps are not multi-view consistent. Previous works [9, 17, 48] simply take mono-depth into a depth-wise loss along with NeRF training. Instead, we propose a novel and effective way to thoroughly integrate mono-depth into our system. First, we explicitly optimise scale and shift parameters for each mono-depth map during NeRF training by penalising the difference between rendered depth and mono-depth. Since NeRF by itself is trained based on multiview consistency, this step transforms mono-depth maps to undistorted multiview consistent depth maps. We further leverage these multiview consistent depth maps in two loss terms: a) a Chamfer Distance loss between two depth maps of adjacent images, which injects relative pose to our system; and b) a depth-based surface rendering loss, which further improves relative pose estimation.

In summary, we propose a method to jointly optimise camera poses and a NeRF from a sequence of images with large camera motion. Our system is enabled by three contributions. **First**, we propose a novel way to integrate mono-depth into unposed-NeRF training by explicitly modelling scale and shift distortions. **Second**, we supply relative poses to the camera-NeRF joint optimisation via an inter-frame loss using undistorted mono-depth maps. **Third**, we further regularise our relative pose estimation with a depth-based surface rendering loss.

As a result, our method is able to handle large camera motion, and outperforms state-of-the-art methods by a significant margin in terms of novel view synthesis quality and camera trajectory accuracy.

2. Related Work

Novel View Synthesis. While early Novel View Synthesis (NVS) approaches applied interpolations between pixels [3], later works often rendered images from 3D reconstruction [1, 6]. In recent years, different representations of the 3D scene are used, *e.g.*meshes [30, 31], Multi-Plane Images [42, 60], layered depth [43] *etc*. Among them, NeRF [24] has become a popular scene representation for its photorealistic rendering.

A number of techniques are proposed to improve NeRF's performance with additional regularisation [13, 26, 55], depth priors [7, 32, 48, 54], surface enhancements [27, 44, 50] or latent codes [41, 45, 53]. Other works [2, 10, 25, 34] have also accelerated NeRF training and rendering. However, most of these approaches require pre-computed camera parameters obtained from SfM algorithms [11, 35].

NeRF With Pose Optimisation. Removing camera parameter preprocessing is an active line of research recently. One category of the methods [33, 39, 61] use a SLAM-style pipeline, that either requires RGB-D inputs or relies on accurate camera poses generated from the SLAM tracking system. Another category of works optimises camera poses with the NeRF model directly. We term this type of method as *unposed-NeRF* in this paper. iNeRF [51] shows that poses for novel view images can be estimated using a reconstructed NeRF model. GNeRF [21] combines Generative Adversarial Networks with NeRF to estimate camera poses but requires a known sampling distribution for poses. More relevant to our work, NeRFmm [47] jointly optimises both camera intrinsics and extrinsics alongside NeRF training. BARF [18] proposes a coarse-to-fine positional encoding strategy for camera poses and NeRF joint optimisation. SC-NeRF [12] further parameterises camera distortion and employs a geometric loss to regularise rays. GARF [4] shows that using Gaussian-MLPs makes joint pose and scene optimisation easier and more accurate. Recently, SiNeRF [49] uses SIREN [37] layers and a novel sampling strategy to alleviate the sub-optimality of joint optimisation in NeRFmm. Although showing promising results on the forward-facing dataset like LLFF [23], these approaches face difficulties when handling challenging camera trajectories with large camera motion. We address this issue by closely integrating mono-depth maps with the joint optimisation of camera parameters and NeRF.

3. Method

We tackle the challenge of handling large camera motions in unposed-NeRF training. Given a sequence of images, camera intrinsics, and their mono-depth estimations, our method recovers camera poses and optimises a NeRF simultaneously. We assume camera intrinsics are available in the image meta block, and we run an off-the-shelf mono-depth network DPT [7] to acquire mono-depth estimations. Without repeating the benefit of mono-depth, we unroll this section around the effective integration of monocular depth into unposed-NeRF training.

The training is a joint optimisation of the NeRF, camera poses, and distortion parameters of each mono-depth

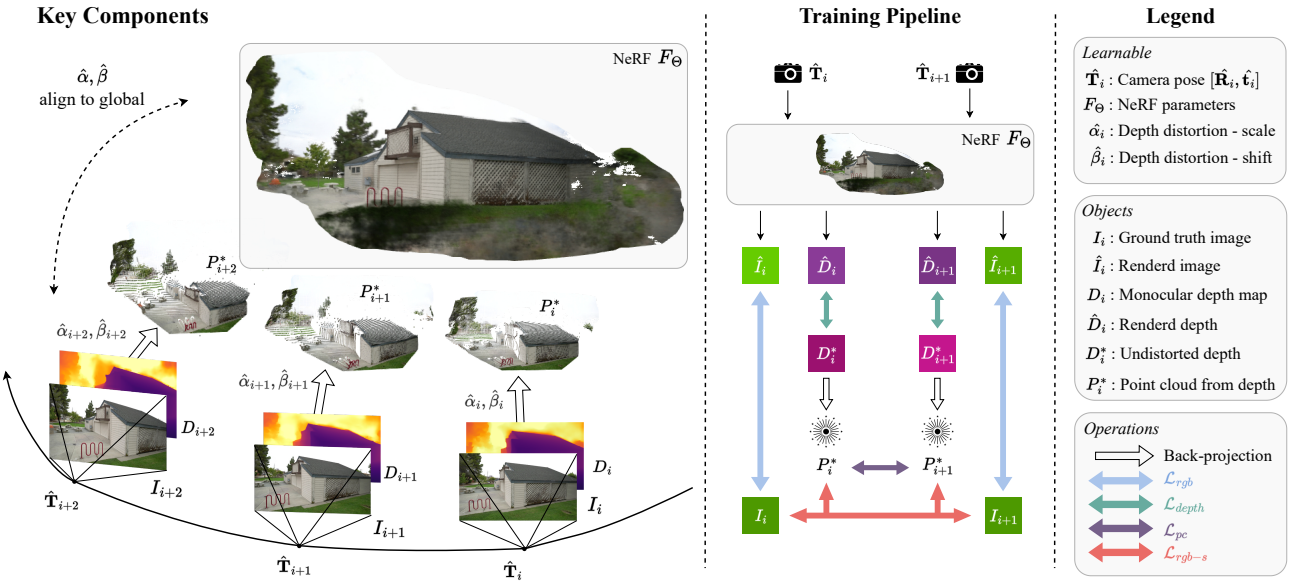


Figure 2. **Method Overview.** Our method takes a sequence of images as input to reconstruct NeRF and jointly estimates the camera poses of the frames. We first generate monocular depth maps from a mono-depth estimation network and reconstruct the point clouds. We then optimise NeRF, camera poses, and depth distortion parameters jointly with inter-frame and NeRF losses.

map. The distortion parameters are supervised by minimising the discrepancies between the mono-depth maps and depth maps rendered from the NeRF, which are multiview consistent. The undistorted depth maps in return effectively mediate the *shape-radiance* ambiguity, which eases the training of NeRF and camera poses.

Specifically, the undistorted depth maps enable two constraints. We constrain global pose estimation by supplying relative pose between adjacent images. This is achieved via a Chamfer-Distance-based correspondence between two point clouds, back-projected from undistorted depth maps. Further, we regularise relative pose estimation with a surface-based photometric consistency where we treat undistorted depth as surface.

We detail our method in the following sections, starting from NeRF in Sec. 3.1 and unposed-NeRF training in Sec. 3.2, looking into mono-depth distortions in Sec. 3.3, followed by our mono-depth enabled loss terms in Sec. 3.4, and finishing with an overall training pipeline Sec. 3.5.

3.1. NeRF

Neural Radiance Field (NeRF) [24] represents a scene as a mapping function $F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$ that maps a 3D location $\mathbf{x} \in \mathbb{R}^3$ and a viewing direction $\mathbf{d} \in \mathbb{R}^3$ to a radiance colour $\mathbf{c} \in \mathbb{R}^3$ and a volume density value σ. This mapping is usually implemented with a neural network parameterised by F_Θ. Given N images $\mathcal{I} = \{I_i \mid i = 0 \dots N-1\}$ with their camera poses $\Pi = \{\pi_i \mid i = 0 \dots N-1\}$, NeRF can be optimised by minimising photometric error $\mathcal{L}_{rgb} = \sum_i^N \|I_i - \hat{I}_i\|_2^2$ between synthesised images $\hat{\mathcal{I}}$ and captured images $\mathcal{I}$:

$$\Theta^* = \arg\min_\Theta \mathcal{L}_{rgb}(\hat{\mathcal{I}} \mid \mathcal{I}, \Pi), \tag{1}$$

where $\hat{I}_i$ is rendered by aggregating radiance colour on camera rays $\mathbf{r}(h) = \mathbf{o} + h\mathbf{d}$ between near and far bound h_n and h_f. More concretely, we synthesise $\hat{I}_i$ with a volumetric rendering function

$$\hat{I}_i(\mathbf{r}) = \int_{h_n}^{h_f} T(h)\sigma(\mathbf{r}(h))\mathbf{c}(\mathbf{r}(h), \mathbf{d})dh, \tag{2}$$

where $T(h) = \exp(-\int_{h_n}^{h} \sigma(\mathbf{r}(s))ds)$ is the accumulated transmittance along a ray. We refer to [24] for further details.

3.2. Joint Optimisation of Poses and NeRF

Prior works [12, 18, 47] show that it is possible to estimate both camera parameters and a NeRF at the same time by minimising the above photometric error $\mathcal{L}_{rgb}$ under the same volumetric rendering process in Eq. (2).

The key lies in conditioning camera ray casting on variable camera parameters Π, as the camera ray $\mathbf{r}$ is a function of camera pose. Mathematically, this joint optimisation can be formulated as:

$$\Theta^*, \Pi^* = \arg\min_{\Theta,\Pi} \mathcal{L}_{rgb}(\hat{\mathcal{I}}, \hat{\Pi} \mid \mathcal{I}), \tag{3}$$

where $\hat{\Pi}$ denotes camera parameters that are updated during optimising. Note that the only difference between Eq. (1)

and Eq. (3) is that Eq. (3) considers camera parameters as variables.

In general, the camera parameters Π include camera intrinsics, poses, and lens distortions. We only consider estimating camera poses in this work, *e.g.*, camera pose for frame I_i is a transformation $\mathbf{T}_i = [\mathbf{R}_i \mid \mathbf{t}_i]$ with a rotation $\mathbf{R}_i \in \mathrm{SO}(3)$ and a translation $\mathbf{t}_i \in \mathbb{R}^3$.

3.3. Undistortion of Monocular Depth

With an off-the-shelf monocular depth network, *e.g.*, DPT [28], we generate mono-depth sequence $\mathcal{D} = \{D_i \mid i = 0 \dots N-1\}$ from input images. Without surprise, mono-depth maps are not multi-view consistent so we aim to recover a sequence of multi-view consistent depth maps, which are further leveraged in our relative pose loss terms.

Specifically, we consider two linear transformation parameters for each mono-depth map, resulting in a sequence of transformation parameters for all frames $\Psi = \{(\alpha_i, \beta_i) \mid i = 0 \dots N-1\}$, where α_i and β_i denote a scale and a shift factor. With multi-view consistent constraint from NeRF, we aim to recover a multi-view consistent depth map D_i^* for D_i:

$$D_i^* = \alpha_i D_i + \beta_i, \tag{4}$$

by joint optimising α_i and β_i along with a NeRF. This joint optimisation is mostly achieved by enforcing the consistency between an undistorted depth map D_i^* and a NeRF rendered depth map $\hat{D}_i$ via a depth loss:

$$\mathcal{L}_{depth} = \sum_{i}^{N} \left\| D_i^* - \hat{D}_i \right\|, \tag{5}$$

where

$$\hat{D}_i(\mathbf{r}) = \int_{h_n}^{h_f} T(h)\, \sigma(\mathbf{r}(h))\, h\, dh \tag{6}$$

denotes a volumetric rendered depth map from NeRF.

It is important to note that both NeRF and mono-depth benefit from Eq. (5). On the one hand, mono-depth provides strong geometry prior for NeRF training, reducing *shape-radiance* ambiguity. On the other hand, NeRF provides multi-view consistency so we can recover a set of multi-view consistent depth maps for relative pose estimations.

3.4. Relative Pose Constraint

Aforementioned unposed-NeRF methods [12, 18, 47] optimise each camera pose independently, resulting in an overfit to target images with incorrect poses. Penalising incorrect relative poses between frames can help to regularise the joint optimisation towards smooth convergence, especially in a complex camera trajectory. Therefore, we propose two losses that constrain relative poses.

Point Cloud Loss. We back-project the undistorted depth maps $\mathcal{D}^*$ using the known camera intrinsics, to point clouds $\mathcal{P}^* = \{P_i^* \mid i = 0 \dots N-1\}$ and optimise the relative pose between consecutive point clouds by minimising a point cloud loss $\mathcal{L}_{pc}$:

$$\mathcal{L}_{pc} = \sum_{(i,j)} l_{cd}(P_j^*, \mathbf{T}_{ji} P_i^*), \tag{7}$$

where $\mathbf{T}_{ji} = \mathbf{T}_j \mathbf{T}_i^{-1}$ represents the related pose that transforms point cloud P_i^* to P_j^*, tuple (i, j) denotes indices of a consecutive pair of instances, and l_{cd} denotes Chamfer Distance:

$$l_{cd}(P_i, P_j) = \sum_{p_i \in P_i} \min_{p_j \in P_j} \|p_i - p_j\|_2 + \sum_{p_j \in P_j} \min_{p_i \in P_i} \|p_i - p_j\|_2. \tag{8}$$

Surface-based Photometric Loss. While the point cloud loss $\mathcal{L}_{pc}$ offers supervision in terms of 3D-3D matching, we observe that a surface-based photometric error can alleviate incorrect matching. With the photometric consistency assumption, this photometric error penalises the differences in appearance between associated pixels. The association is established by projecting the point cloud P_i^* onto images I_i and I_j.

The surface-based photometric loss can then be defined as:

$$\mathcal{L}_{rgb-s} = \sum_{(i,j)} \| I_i \langle \mathbf{K}_i P_i^* \rangle - I_j \langle \mathbf{K}_j \mathbf{T}_j \mathbf{T}_i^{-1} P_i^* \rangle \|, \tag{9}$$

where $\langle \cdot \rangle$ represents the sampling operation on the image and $\mathbf{K}_i$ denotes a projection matrix for i_{th} camera.

3.5. Overall Training Pipeline

Assembling all loss terms, we get the overall loss function:

$$\mathcal{L} = \mathcal{L}_{rgb} + \lambda_1 \mathcal{L}_{depth} + \lambda_2 \mathcal{L}_{pc} + \lambda_3 \mathcal{L}_{rgb-s}, \tag{10}$$

where $\lambda_1, \lambda_2, \lambda_3$ are the weighting factors for respective loss terms. By minimising the combined of loss $\mathcal{L}$:

$$\Theta^*, \Pi^*, \Psi^* = \arg\min_{\Theta, \Pi, \Psi} \mathcal{L}(\hat{\mathcal{I}}, \hat{\mathcal{D}}, \hat{\Pi}, \hat{\Psi} \mid \mathcal{I}, \mathcal{D}), \tag{11}$$

our method returns the optimised NeRF parameters Θ, camera poses Π, and distortion parameters Ψ.

4. Experiments

We begin with a description of our experimental setup in Sec. 4.1. In Sec. 4.2, we compare our method with pose-unknown methods. Next, we compare our method with the COLMAP-assisted NeRF baseline in Sec. 4.3. Lastly, we conduct ablation studies in Sec. 4.4.



(a) BARF (b) NeRFmm (c) SC-NeRF (d) Ours (e) Ground Truth

Figure 3. **Qualitative results of novel view synthesis and depth prediction on Tanks and Temples.** We visualise the synthesised images and the rendered depth maps (top left of each image) for all methods. NoPe-NeRF is able to recover details for both colour and geometry.

4.1. Experimental Setup

Datasets. We conduct experiments on two datasets *Tanks and Temples* [15] and *ScanNet* [5]. **Tanks and Temples**: we use 8 scenes to evaluate pose accuracy and novel view synthesis quality. We chose scenes captured at both indoor and outdoor locations, with different frame sampling rates and lengths. All images are down-sampled to a resolution of 960×540. For the *family* scene, we sample 200 images and take 100 frames with odd frame ids as training images and the remaining 100 frames for novel view synthesis, in order to analyse the performance under smooth motion. For the remaining scenes, following NeRF [24], 1/8 of the images in each sequence are held out for novel view synthesis, unless otherwise specified. **ScanNet**: we select 4 scenes for evaluating pose accuracy, depth accuracy, and novel view synthesis quality. For each scene, we take 80-100 consecutive images and use 1/8 of these images for novel view synthesis. For evaluation, we employ depth maps and poses provided by ScanNet as ground truth. ScanNet images are down-sampled to 648×484. We crop images with dark orders during preprocessing.

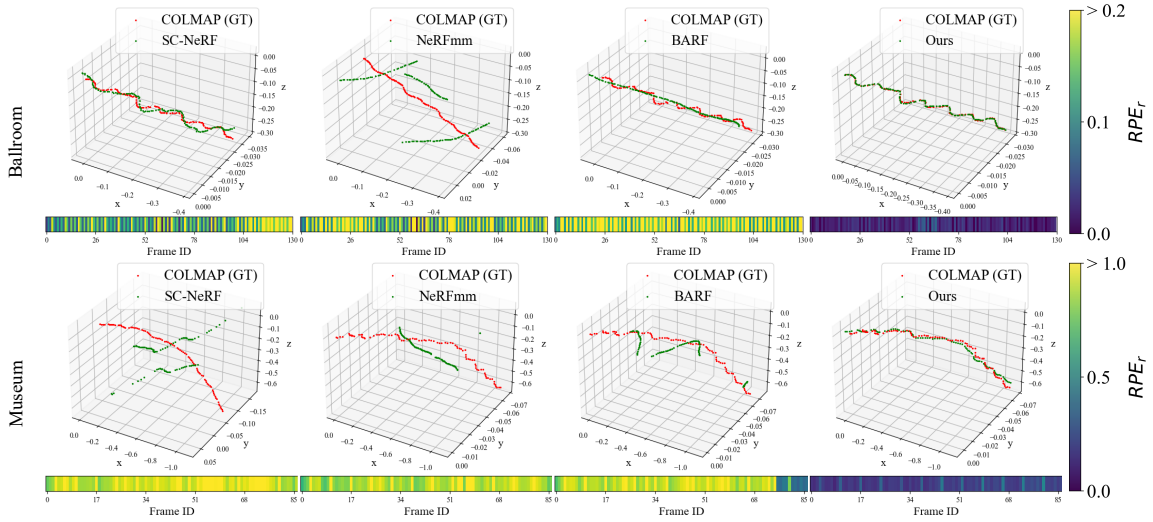
Metrics. We evaluate our proposed method in three aspects. For **novel view synthesis**, we follow previous methods [12,18,47], and use standard evaluation metrics, including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [46] and Learned Perceptual Image Patch Similarity (LPIPS) [57]. For **pose** evaluation, We use standard visual odometry metrics [16, 38, 58], including the Absolute Trajectory Error (ATE) and Relative Pose Error (RPE). ATE measures the difference between the estimated camera positions and the ground truth positions. RPE measures the relative pose errors between pairs of images, which consists of relative rotation error (RPE_r) and relative translation error (RPE_t). The estimated trajectory is aligned with the ground truth using Sim(3) with 7 degrees of freedom. We use standard depth metrics [8,19,20,40] (Abs Rel, Sq Rel, RMSE, RMSE log, δ_1, δ_2 and δ_3) for **depth** evaluation. For further detail, please refer to the supplementary material. To recover the metric scale, we follow Zhou *et al*. [59] and match the median value between rendered and ground truth depth maps.

Implementation Details. Our model architecture is based on NeRF [24] with a few modifications: a) replacing ReLU activation function with Softplus and b) sampling 128 points along each ray uniformly with noise, between a predefined range $(0.1, 10)$. We use 2 separate Adam optimisers [14] for NeRF and other parameters. The initial learning rate for NeRF is 0.001 and for the pose and distortion is 0.0005. Camera rotations are optimised in axis-angle representation $\boldsymbol{\phi}_i \in \mathfrak{so}(3)$. We first train the model with all losses with constant learning rates until the inter-frame losses converge. Then, we remove the inter-frame losses and depth loss to refine the model with the RGB loss only. We decay the learning rates with different schedulers to refine for 10,000 epochs. We balance the loss terms with $\lambda_1 = 0.04$, $\lambda_2 = 1.0$ and $\lambda_3 = 1.0$. For each training step, we randomly sample 1024 pixels (rays) from each input image and 128 samples per ray. More details are provided in the supplementary material.

4.2. Comparing With Pose-Unknown Methods

We compare our method with pose-unknown baselines, including BARF [18], NeRFmm [47] and SC-NeRF [12].

View Synthesis Quality. To obtain the camera poses of



(a) BARF (b) NeRFmm (c) SC-NeRF (d) Ours

Figure 4. **Pose Estimation Comparison.** We visualise the trajectory (3D plot) and relative rotation errors RPE_r (bottom colour bar) of each method on *Ballroom* and *Museum*. The colour bar on the right shows the relative scaling of colour. More results are in the supplementary.

	scenes	Ours			BARF			NeRFmm			SC-NeRF		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
ScanNet	0079_00	**32.47**	**0.84**	**0.41**	32.31	0.83	0.43	30.59	0.81	0.49	31.33	0.82	0.46
	0418_00	**31.33**	**0.79**	**0.34**	31.24	**0.79**	0.35	30.00	0.77	0.40	29.05	0.75	0.43
	0301_00	**29.83**	**0.77**	0.36	29.31	0.76	0.38	27.84	0.72	0.45	29.45	**0.77**	**0.35**
	0431_00	**33.83**	**0.91**	**0.39**	32.77	0.90	0.41	31.44	0.88	0.45	32.57	0.90	0.40
	mean	**31.86**	**0.83**	**0.38**	31.41	0.82	0.39	29.97	0.80	0.45	30.60	0.81	0.41
Tanks and Temples	Church	**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	**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	**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	**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	**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	**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	**29.48**	**0.80**	**0.38**	25.85	0.69	0.57	25.40	00.69	0.52	26.46	0.73	0.49
	Ignatius	**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	**26.34**	**0.74**	**0.39**	23.42	0.61	0.54	22.50	0.59	0.54	23.76	0.65	0.48

Table 1. **Novel view synthesis results on ScanNet and Tanks and Temples**. Each baseline method is trained with its public code under the original settings and evaluated with the same evaluation protocol.

test views for rendering, we minimise the photometric error of the synthesised images while keeping the NeRF model fixed, as in NeRFmm [47]. Each test pose is initialised with the learned pose of the training frame that is closest to it. We use the same pre-processing for all baseline approaches, which results in higher accuracy than their original implementations. More details are provided in the supplementary material. Our method outperforms all the baselines by a large margin. The quantitative results are summarised in Tab. 1, and qualitative results are shown in Fig. 3.

We recognised that because the test views, which are sampled from videos, are close to the training views, good results may be obtained due to overfitting to the training images. Therefore, we conduct an additional qualitative evaluation on more novel views. Specifically, we fit a bezier curve from the estimated training poses and sample interpolated poses for each method to render novel view videos. Sampled results are shown in Fig. 5, and the rendered videos are in the supplementary material. These results show that our method renders photo-realistic images consistently, while other methods generate visible artifacts.

Camera Pose. Our method significantly outperforms other baselines in all metrics. The quantitative pose evaluation results are shown in Tab. 2. For ScanNet, we use the camera poses provided by the dataset as ground truth. For Tanks and Temples, not every video comes with ground truth poses, so we use COLMAP estimations for reference. Our estimated trajectory is better aligned with the ground

	scenes	Ours			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
ScanNet	0079_00	**0.752**	**0.204**	**0.023**	1.110	0.480	0.062	1.706	0.636	0.100	2.064	0.664	0.115
	0418_00	**0.455**	**0.119**	**0.015**	1.398	0.538	0.020	1.402	0.460	0.013	1.528	0.502	0.016
	0301_00	**0.399**	**0.123**	**0.013**	1.316	0.777	0.219	3.097	0.894	0.288	1.133	0.422	0.056
	0431_00	**1.625**	**0.274**	**0.069**	6.024	0.754	0.168	6.799	0.624	0.496	4.110	0.499	0.205
	mean	**0.808**	**0.180**	**0.030**	2.462	0.637	0.117	3.251	0.654	0.224	2.209	0.522	0.098
Tanks and Temples	Church	**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	**0.046**	**0.032**	**0.004**	0.314	0.265	0.050	1.629	0.494	0.159	1.317	0.429	0.157
	Museum	**0.207**	**0.202**	**0.020**	3.442	1.128	0.263	4.134	1.051	0.346	8.339	1.491	0.316
	Family	**0.047**	**0.015**	**0.001**	1.371	0.591	0.115	2.743	0.537	0.120	1.171	0.499	0.142
	Horse	**0.179**	**0.017**	**0.003**	1.333	0.394	0.014	1.349	0.434	0.018	1.366	0.438	0.019
	Ballroom	**0.041**	**0.018**	**0.002**	0.531	0.228	0.018	0.449	0.177	0.031	0.328	0.146	0.012
	Francis	**0.057**	**0.009**	**0.005**	1.321	0.558	0.082	1.647	0.618	0.207	1.233	0.483	0.192
	Ignatius	**0.026**	**0.005**	**0.002**	0.736	0.324	0.029	1.302	0.379	0.041	0.533	0.240	0.085
	mean	**0.080**	**0.038**	**0.006**	1.046	0.441	0.078	1.735	0.477	0.123	1.890	0.489	0.129

Table 2. **Pose accuracy on ScanNet and 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.

	Abs Rel $\downarrow$	Sq Rel $\downarrow$	RMSE $\downarrow$	RMSE log $\downarrow$	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$
Ours	**0.141**	**0.137**	**0.568**	**0.176**	**0.828**	**0.970**	**0.987**
BARF	0.376	0.684	0.990	0.401	0.490	0.751	0.884
NeRFmm	0.590	1.721	1.672	0.587	0.316	0.560	0.743
SC-NeRF	0.417	0.642	1.079	0.476	0.362	0.658	0.832
DPT	0.197	0.246	0.751	0.226	0.747	0.934	0.975

Table 3. **Depth map evaluation on ScanNet**. Our depth estimation is more accurate than baseline models BARF [18], NeRFmm [47] and SC-NeRF [12]. Compared with DPT [59], we show our depth is more accurate after undistortion.

truth than other methods, and our estimated rotation is two orders of magnitudes more accurate than others. We visualise the camera trajectories and rotations in Fig. 4.

Depth. We evaluate the accuracy of the rendered depth maps on ScanNet, which provides the ground-truth depths for evaluation. Our rendered depth maps achieve superior accuracy over the previous alternatives. We also compare with the mono-depth maps estimated by DPT. Our rendered depth maps, after undistortion using multiview consistency in the NeRF optimisation, outperform DPT by a large margin. The results are summarised in Tab. 3, and sampled qualitative results are illustrated in Fig. 3.

4.3. Comparing With COLMAP Assisted NeRF

We make a comparison of pose estimation accuracy between our method and COLMAP against ground truth poses in ScanNet. We achieve on-par accuracy with COLMAP, as shown in Tab. 4. We further analyse the novel view synthesis quality of the NeRF model trained with our learned poses to COLMAP poses on ScanNet and Tanks and Temples. The original NeRF training contains two stages, finding poses using COLMAP and optimising the scene representation. In order to make our comparison fairer, in this section only, we mimic a similar two-stage training as the original NeRF [24]. In the first stage, we train our method with all losses for camera pose estimation, *i.e.*, mimicking the COLMAP processing. Then, we fix the optimised poses

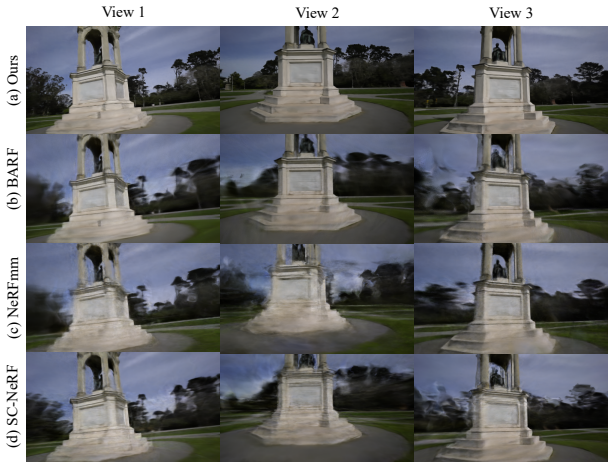


Figure 5. **Sampled frames from rendered novel view videos.** For each method, we fit the learned trajectory with a bezier curve and uniformly sample new viewpoints for rendering. Our method generates significantly better results than previous methods, which show visible artifacts. The full rendered videos and details about generating novel views are provided in the supplementary.

and train a NeRF model from scratch, using the same settings and loss as the original NeRF. This evaluation enables us to compare our estimated poses to the COLMAP poses indirectly, *i.e.*, in terms of contribution to view synthesis.

Our two-stage method outperforms the COLMAP-assisted NeRF baseline, which indicates a better pose estimation for novel view synthesis. The results are summarised in Tab. 5.

As is commonly known, COLMAP performs poorly in low-texture scenes and sometimes fails to find accurate camera poses. Fig. 6 shows an example of a low-texture scene where COLMAP provides inaccurate pose estimation that causes NeRF to render images with visible artifacts. In contrast, our method renders high-quality images, thanks to robust optimisation of camera pose.

scenes	Ours			COLMAP		
	RPE$_t$ ↓	RPE$_r$ ↓	ATE↓	RPE$_t$	RPE$_r$	ATE
0079_00	0.752	**0.204**	0.023	**0.655**	0.221	**0.012**
0418_00	**0.455**	**0.119**	**0.015**	0.491	0.124	0.016
0301_00	**0.399**	**0.123**	0.013	0.414	0.136	**0.009**
0431_00	1.625	0.274	0.069	**1.292**	**0.249**	**0.051**
mean	0.808	**0.180**	0.030	**0.713**	0.182	**0.022**

Table 4. **Comparison of pose accuracy with COLMAP on ScanNet.**



(a) COLMAP+NeRF (b) Ours (c) Ground Truth

Figure 6. **COLMAP failure case.** On a rotation-dominant sequence with low-texture areas, COLMAP fails to estimate correct poses, which results in artifacts in synthesised images.

	scenes	Ours			Ours-r			COLMAP+NeRF		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
ScanNet	0079_00	32.47	0.84	0.41	**33.12**	**0.85**	**0.40**	31.98	0.83	0.43
	0418_00	**31.33**	**0.79**	**0.34**	30.49	0.77	0.40	30.60	0.78	0.40
	0301_00	29.83	0.77	0.36	**30.05**	**0.78**	**0.34**	30.01	**0.78**	0.36
	0431_00	33.83	0.91	0.39	**33.86**	**0.91**	**0.39**	33.54	**0.91**	**0.39**
	mean	31.86	**0.83**	**0.38**	**31.88**	**0.83**	**0.38**	31.53	0.82	0.40
Tanks and Temples	Church	25.17	0.73	0.39	**26.74**	**0.78**	**0.32**	25.72	0.75	0.37
	Barn	26.35	0.69	0.44	26.58	**0.71**	**0.42**	**26.72**	**0.71**	**0.42**
	Museum	26.77	0.76	0.35	26.98	0.77	0.36	**27.21**	**0.78**	**0.34**
	Family	26.01	0.74	0.41	26.21	0.75	0.40	**26.61**	**0.77**	**0.39**
	Horse	27.64	**0.84**	**0.26**	**28.06**	**0.84**	**0.26**	27.02	0.82	0.29
	Ballroom	25.33	0.72	**0.38**	**25.53**	**0.73**	**0.38**	25.47	**0.73**	**0.38**
	Francis	29.48	0.80	**0.38**	29.73	**0.81**	**0.38**	30.05	**0.81**	**0.38**
	Ignatius	23.96	0.61	0.47	23.98	**0.62**	**0.46**	24.08	0.61	0.47
	mean	26.34	0.74	0.39	**26.73**	**0.75**	**0.37**	26.61	**0.75**	0.38

Table 5. **Comparison to NeRF with COLMAP poses.** Our two-stage method (Ours-r) outperforms both COLMAP+NeRF and our one-stage method (Ours).

Interestingly, this experiment also reveals that the two-stage method shows higher accuracy than the one-stage method. We hypothesise that the joint optimisation (from randomly initialised poses) in the one-stage approach causes the NeRF optimisation to be trapped in a local minimum, potentially due to the bad pose initialisation. The two-stage approach circumvents this issue by re-initialising the NeRF and re-training with well-optimised poses, resulting in higher performance.

4.4. Ablation Study

In this section, we analyse the effectiveness of the parameters and components that have been added to our model. The results of ablation studies are shown in Tab. 6.

Effect of Distortion Parameters. We find that ignoring depth distortions (*i.e.*, setting scales to 1 and shifts to 0 as constants) leads to a degradation in pose accuracy, as inconsistent distortions of depth maps introduce errors to the estimation of relative poses and confuse NeRF for geometry reconstruction.

	NVS			Pose		
	PSNR↑	SSIM ↑	LPIPS ↓	RPE$_t$ ↓	RPE$_r$ ↓	ATE ↓
Ours	**31.86**	**0.83**	**0.38**	**0.801**	**0.181**	**0.031**
Ours w/o α, β	31.46	0.82	0.39	1.929	0.321	0.066
Ours w/o L_{pc}	31.73	0.82	**0.38**	2.227	0.453	0.101
Ours w/o L_{rgb-s}	31.05	0.81	0.41	1.814	0.401	0.156
Ours w/o L_{depth}	31.20	0.81	0.40	1.498	0.383	0.089

Table 6. **Ablation study results on ScanNet.**

Effect of Inter-frame Losses. We observe that the inter-frame losses are the major contributor to improving relative poses. When removing the pairwise point cloud loss L_{pc} or the surface-based photometric loss L_{rgb-s}, there is less constraint between frames, and thus the pose accuracy becomes lower.

Effect of NeRF Losses. When the depth loss L_{depth} is removed, the distortions of input depth maps are only optimised locally through the inter-frame losses. We find that this can lead to drift and degradation in pose accuracy.

4.5. Limitations

Our proposed method optimises camera pose and the NeRF model jointly and works on challenging scenes where other baselines fail. However, the optimisation of the model is also affected by non-linear distortions and the accuracy of the mono-depth estimation, which we did not consider.

5. Conclusion

In this work, we present NoPe-NeRF, an end-to-end differentiable model for joint camera pose estimation and novel view synthesis from a sequence of images. We demonstrate that previous approaches have difficulty with complex trajectories. To tackle this challenge, we use mono-depth maps to constrain the relative poses between frames and regularise the geometry of NeRF, which leads to better pose estimation. We show the effectiveness and robustness of NoPe-NeRF on challenging scenes. The improved pose estimation leads to better novel view synthesis quality and geometry reconstruction compared with other approaches. We believe our method is an important step towards applying the unknown-pose NeRF models to large-scale scenes in the future.

Acknowledgements

We thank Theo Costain, Michael Hobley, Shuai Chen and Xinghui Li for their helpful proofreading and discussions. Wenjing Bian is supported by the China Scholarship Council - University of Oxford Scholarship.

References

[1] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *SIGGRAPH*, 2001. 2

[2] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, 2022. 2

[3] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *SIGGRAPH*, 1993. 2

[4] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. Garf: Gaussian activated radiance fields for high fidelity reconstruction and pose estimation. *arXiv e-prints*, 2022. 2

[5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 5, 11

[6] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *SIGGRAPH*, 1996. 2

[7] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *CVPR*, 2022. 2

[8] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018. 5

[9] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *ICCV*, 2021. 2

[10] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *ICCV*, 2021. 2

[11] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. 2003. 2

[12] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *ICCV*, 2021. 1, 2, 3, 4, 5, 7, 11

[13] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *CVPR*, 2022. 2

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015. 5

[15] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 2017. 5, 11

[16] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *CVPR*, 2021. 5

[17] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021. 2

[18] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021. 1, 2, 3, 4, 5, 7, 11

[19] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *TPAMI*, 2015. 5

[20] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ToG*. 5

[21] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. Gnerf: Gan-based neural radiance field without posed camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 2

[22] S Mahdi H Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yagiz Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *CVPR*, 2021. 2

[23] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019. 2, 12

[24] 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. 1, 2, 3, 5, 7

[25] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.* 2

[26] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *CVPR*, 2022. 2

[27] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *ICCV*, 2021. 2

[28] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 2, 4

[29] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *TPAMI*, 2020. 2

[30] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *ECCV*, 2020. 2

[31] Gernot Riegler and Vladlen Koltun. Stable view synthesis. In *CVPR*, 2021. 2

[32] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *CVPR*, 2022. 2

[33] Antoni Rosinol, John J Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. *arXiv preprint arXiv:2210.13641*, 2022. 2

[34] Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 2

[35] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 2

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 12

[37] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *NeurIPS*, 2020. 2

[38] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IROS*. IEEE, 2012. 5

[39] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew Davison. iMAP: Implicit mapping and positioning in real-time. In *ICCV*, 2021. 2

[40] Libo Sun, Jia-Wang Bian, Huangying Zhan, Wei Yin, Ian Reid, and Chunhua Shen. Sc-depthv3: Robust self-supervised monocular depth estimation for dynamic scenes. *arXiv preprint arXiv:2211.03660*, 2022. 5

[41] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d representation and rendering. In *ICCV*, 2021. 2

[42] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020. 2

[43] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *ECCV*, 2018. 2

[44] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *NeurIPS*, 2021. 2

[45] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 2

[46] 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. 5, 11

[47] 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. 1, 2, 3, 4, 5, 6, 7, 11

[48] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *ICCV*, 2021. 2

[49] Yitong Xia, Hao Tang, Radu Timofte, and Luc Van Gool. Sinerf: Sinusoidal neural radiance fields for joint pose estimation and scene reconstruction. 2022. 2

[50] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *NeurIPS*, 2021. 2

[51] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In *IROS*, 2021. 2

[52] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Simon Chen, Yifan Liu, and Chunhua Shen. Towards accurate reconstruction of 3d scene shape from a single monocular image. *TPAMI*, 2022. 2

[53] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 2

[54] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *NeurIPS*, 2022. 2

[55] Jian Zhang, Yuanqing Zhang, Huan Fu, Xiaowei Zhou, Bowen Cai, Jinchi Huang, Rongfei Jia, Binqiang Zhao, and Xing Tang. Ray priors through reprojection: Improving neural radiance fields for novel view extrapolation. In *CVPR*, 2022. 2

[56] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2

[57] 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. 5, 12

[58] Zichao Zhang and Davide Scaramuzza. A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry. In *IROS*. IEEE, 2018. 5, 11

[59] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 5, 7

[60] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. 2018. 2

[61] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *CVPR*, 2022. 2

Appendix A. Implementation Details

The following sections include more details about the datasets we use, our training procedure and evaluation metrics.

A.1. Dataset

We select sequences containing dramatic camera motions from ScanNet [5] and Tanks and Tamples [15] for training and evaluation. Tab. 7 lists details about these sequences, where *Max rotation* denotes the maximum relative rotation angle between any two frames in a sequence. The sampled images are further split into training and test sets. Starting from the 5*th* image, we sample every 8*th* image in a sequence as a test image. However, this leads to a change in the sampling rate in the temporal domain among training images. We found that the rotation errors are often higher than average at these positions where the sampling rate changes. In order to study the effect of the sampling rate changes, for scene *Family* in Tanks and Temples [15], we sample every other image as test images, i.e. training on images with odd frame ids and testing on images with even frame ids.

	Scenes	Type	Seq. length	Frame rate	Max. rotation (deg)
ScanNet	0079_00	indoor	90	30	54.4
	0418_00	indoor	80	30	27.5
	0301_00	indoor	100	30	43.7
	0431_00	indoor	100	30	45.8
Tanks and Temples	Church	indoor	400	30	37.3
	Barn	outdoor	150	10	47.5
	Museum	indoor	100	10	76.2
	Family	outdoor	200	30	35.4
	Horse	outdoor	120	20	39.0
	Ballroom	indoor	150	20	30.3
	Francis	outdoor	150	10	47.5
	Ignatius	outdoor	120	20	26.0

Table 7. **Details of selected sequences.** We downsample several videos to a lower frame rate. FPS denote frame per second. *Max rotation* denotes the maximum relative rotation angle between any two frames in a sequence. We show our method can handle dramatic camera motion (large maximum rotation angle) whereas previous methods can only handle forward-facing scenes.

A.2. Training Details

During training, we sample 1024 pixels/rays for an image and we sample 128 points along each ray for our approaches and all baselines. For all approaches, we use the same pre-defined sampling range (i.e., near and far) and sample uniformly between this range. During refinement, the learning rate of NeRF model decays every 10 epochs with 0.9954, and the learning rate for the camera poses decays every 100 epochs with 0.9. Note that we only use RGB loss during refinement and depth distortion parameters are no longer optimised. As the scene scales can be arbitrary, the optimised scale parameter of the depth map during training is also arbitrary. To avoid scale collapsing (all scales reduced to 0.0) during training, we manually set the scale of the depth map for the last frame to 1.0. We also use the normalised point clouds when computing the inter-frame point cloud loss.

A.3. Test-time Optimisation

During the evaluation for novel view synthesis, following our baselines NeRFmm [47], BARF [18] and SC-NeRF [12], we run a test-time optimisation to align the camera poses of the test set by minimising the photometric error on the synthesised images, while keeping the trained NeRF model froze. Although all these baseline methods have their own way to align camera poses (discussed below), all of them fail to align camera poses in complex camera trajectories in ScanNet and Tanks and Temples.

To fairly evaluate all methods in challenging camera trajectories, we propose to align test camera poses by first initialising from learned poses of adjacent training images, followed by a test-time optimisation. We shorthand this alignment as **Neighbour + opt**. In practice, we find this initialisation is robust and provides the best alignment for all approaches. All results in our main paper are evaluated in this way.

The following paragraphs outline previous alignment methods, and we show a comparison for all method with a ScanNet scene in Tab. 8.

Identity + opt. BARF [18] uses test-time optimisation to identify poses for the test frames, where all poses are initialised with identity matrices. This initialisation works well for simple forward-facing scenes, but not for complex trajectories. The optimisation is sensitive to the learning rate, and can easily fall into local minima when the target pose is far from the identity initialisation.

Sim(3) + opt. In NeRFmm [47], the poses are first initialised using Sim(3) alignment with an ATE toolbox [58]. Then, an additional test-time optimisation is used to further adjust the test poses. This initialisation works well when the learned poses can be aligned precisely to COLMAP poses (Ours in Tab. 8). However, incorrect pose estimations can affect the Sim(3) alignment.

Sim(3) + no opt. In SC-NeRF [12], the test poses are identified using a Sim(3) alignment between COLMAP poses and the learned poses. And no test-time optimisation is used. However, the results are biased toward COLMAP estimations, and misalignment can affect the view synthesis quality significantly.

A.4. Evaluation Metrics

Novel View Synthesis. We use Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [46] and Learned Perceptual Image Patch Similarity

	Sim(3) + no opt.			Identity + opt.			Sim(3) + opt.			(4) Neighbour + opt		
	PSNR↑	SSIM ↑	LPIPS ↓	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Ours	17.24	0.62	0.58	13.38	0.39	0.70	32.47	0.84	0.41	32.47	0.84	0.41
BARF	14.68	0.55	0.66	19.56	0.65	0.57	17.82	0.60	0.61	32.31	0.83	0.43
NeRFmm	11.28	0.40	0.80	30.59	0.81	0.49	12.46	0.43	0.80	30.59	0.81	0.49
SC-NeRF	10.68	0.38	0.80	22.39	0.71	0.55	11.25	0.40	0.80	31.33	0.82	0.46

Table 8. **Comparison of various pose alignment methods during test-time optimisation (ScanNet 0079_00).**

scenes	Ours			NeRFmm		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR	SSIM	LPIPS
Fern	**23.01**	**0.71**	**0.38**	20.58	0.59	0.50
Flower	**29.39**	**0.86**	**0.19**	27.02	0.76	0.32
Fortress	**29.38**	**0.80**	**0.28**	24.94	0.57	0.57
Horns	**25.24**	**0.73**	**0.37**	23.67	0.66	0.48
Leaves	**19.85**	**0.60**	**0.40**	19.46	0.55	0.46
Orchids	**19.51**	**0.56**	**0.43**	16.77	0.40	0.55
Room	**28.54**	**0.89**	**0.28**	26.14	0.84	0.39
Trex	**25.82**	**0.84**	**0.29**	24.13	0.77	0.39
mean	**25.09**	**0.75**	**0.33**	22.84	0.64	0.46

Table 9. **Novel view synthesis results on LLFF-NeRF dataset.**

(LPIPS) [57] to measure the novel view synthesis quality. For LPIPS, we use a VGG architecture [36].

Depth. The error metrics we use for depth evaluation include Abs Rel, Sq Rel, RMSE, RMSE log, δ_1, δ_2 and δ_3. The definitions are as follows:

- Abs Rel: $\frac{1}{|\mathcal{V}|}\sum_{d\in\mathcal{V}}\|d-d_{gt}\|/d_{gt}$;
- Sq Rel: $\frac{1}{|\mathcal{V}|}\sum_{d\in\mathcal{V}}\|d-d_{gt}\|_2^2/d_{gt}$;
- RMSE: $\sqrt{\frac{1}{|\mathcal{V}|}\sum_{d\in\mathcal{V}}\|d-d_{gt}\|_2^2}$;
- RMSE log: $\sqrt{\frac{1}{|\mathcal{V}|}\sum_{d\in\mathcal{V}}\|\log d-\log d_{gt}\|_2^2}$;
- δ_i: % of y s.t. $max(\frac{d}{d_{gt}},\frac{d_{gt}}{d})=\delta<i$;

where d is the estimated depth, d_{gt} is the ground truth depth, and $\mathcal{V}$ is the collection of all valid pixels on a depth map.

Appendix B. Additional Results

LLFF-NeRF Dataset. We compare our approach against NeRFmm on the LLFF-NeRF dataset [23] in terms of novel view synthesis quality (Tab. 9) and pose accuracy (Tab. 10). We show better performances than NeRFmm in both pose accuracy and synthesis quality. We use the normalized device coordinate (NDC) for both approaches.

Depth Estimation. We show detailed depth evaluation results for ScanNet scenes in Tabs. 11 to 14. Our depth estimation accuracy outperforms other baselines by a large margin.

scenes	Ours			NeRFmm		
	RPE_t ↓	RPE_r ↓	ATE↓	RPE_t	RPE_r	ATE
Fern	**0.252**	**0.993**	**0.003**	0.706	1.816	0.007
Flower	**0.035**	**0.096**	**0.001**	0.086	0.418	**0.001**
Fortress	**0.081**	**0.296**	**0.001**	0.233	0.739	0.004
Horns	**0.217**	**0.452**	**0.004**	0.321	0.850	0.008
Leaves	0.218	0.143	0.002	**0.138**	**0.051**	**0.001**
Orchids	**0.203**	**0.383**	**0.003**	0.686	2.030	0.010
Room	**0.244**	**0.936**	**0.004**	0.670	1.664	0.011
Trex	**0.219**	**0.319**	**0.004**	0.542	0.775	0.009
mean	**0.184**	**0.452**	**0.003**	0.423	1.043	0.006

Table 10. **Pose accuracy on LLFF-NeRF dataset.**

0079_00	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Ours	**0.099**	**0.047**	**0.335**	**0.128**	**0.904**	**0.995**	**1.000**
BARF	0.208	0.165	0.588	0.263	0.639	0.896	0.983
NeRFmm	0.494	1.049	1.419	0.534	0.378	0.567	0.765
SC-NeRF	0.360	0.450	0.902	0.396	0.407	0.730	0.908
DPT	0.149	0.095	0.456	0.173	0.818	0.978	0.999

Table 11. **Depth map evaluation on ScanNet 0079_00.**

0418_00	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Ours	**0.152**	**0.137**	**0.645**	**0.185**	**0.738**	**0.988**	**0.997**
BARF	0.718	1.715	1.563	0.630	0.205	0.569	0.769
NeRFmm	0.907	3.650	2.176	0.769	0.240	0.456	0.621
SC-NeRF	0.319	0.441	0.898	0.377	0.456	0.792	0.930
DPT	0.190	0.187	0.745	0.211	0.719	0.965	**0.997**

Table 12. **Depth map evaluation on ScanNet 0418_00.**

0301_00	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Ours	0.185	0.252	0.711	**0.233**	**0.792**	**0.918**	**0.958**
BARF	**0.179**	**0.146**	**0.502**	0.268	0.736	0.883	0.938
NeRFmm	0.444	0.830	1.239	0.481	0.397	0.680	0.845
SC-NeRF	0.383	0.378	0.810	0.452	0.360	0.663	0.846
DPT	0.317	0.568	1.133	0.350	0.597	0.821	0.914

Table 13. **Depth map evaluation on ScanNet 0301_00.**

0431_00	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Ours	**0.127**	**0.111**	**0.579**	**0.160**	**0.877**	**0.978**	**0.994**
BARF	0.398	0.710	1.307	0.444	0.381	0.655	0.847
NeRFmm	0.514	1.354	1.855	0.562	0.250	0.539	0.742
SC-NeRF	0.608	1.300	1.706	0.677	0.225	0.446	0.645
DPT	0.132	0.135	0.670	0.171	0.855	0.973	0.991

Table 14. **Depth map evaluation on ScanNet 0431_00.**

Pose Estimation. We visualise additional results for pose estimation on Tanks and Temples (Fig. 9) and ScanNet (Fig. 10).



(a) BARF (b) NeRFmm (c) SC-NeRF (d) Ours (e) Ground Truth

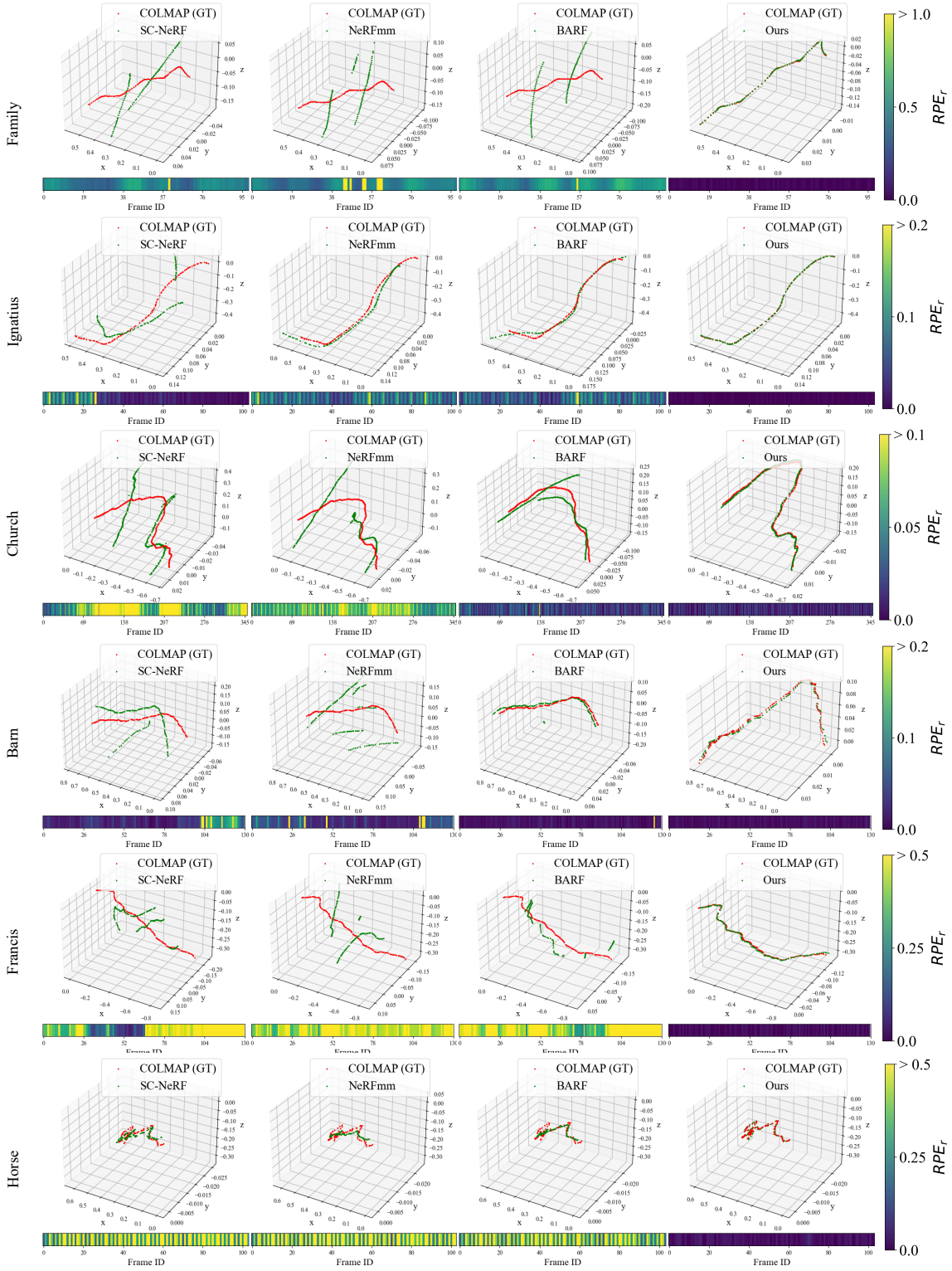
Figure 7. **Qualitative results of novel view synthesis and depth prediction on Tanks and Temples.** We visualise the synthesised images and the rendered depth maps (top left of each image) for all methods. NoPe-NeRF is able to recover details for both colour and geometry.

More Visualisations. We present additional qualitative results for novel view synthesis and depth estimation on Tanks and Temples (Fig. 7) and ScanNet (Fig. 8).



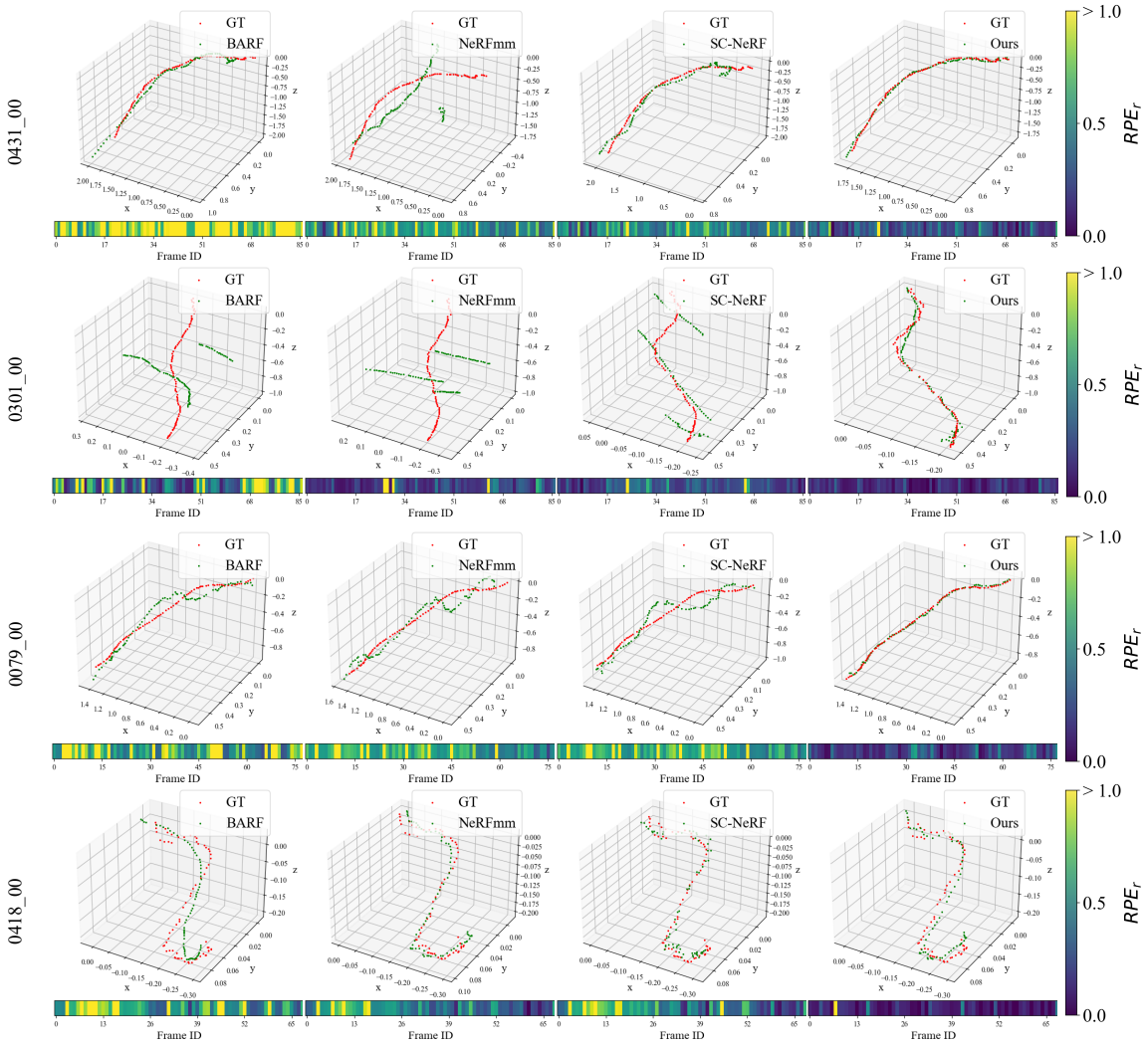
(a) BARF (b) NeRFmm (c) SC-NeRF (d) Ours (e) Ground Truth

Figure 8. **Qualitative results of novel view synthesis and depth prediction on ScanNet.** We visualise the synthesised images and the rendered depth maps (top left of each image) for all methods. NoPe-NeRF is able to recover details for both colour and geometry.



(a) BARF (b) NeRFmm (c) SC-NeRF (d) Ours

Figure 9. **Pose Estimation Comparison on Tanks and Temples.** We visualise the trajectory (3D plot) and relative rotation errors RPE_r (bottom colour bar) of each method on *Ballroom* and *Museum*. The colour bar on the right shows the relative scaling of colour.



(a) BARF (b) NeRFmm (c) SC-NeRF (d) Ours

Figure 10. **Pose Estimation Comparison on ScanNet.** We visualise the trajectory (3D plot) and relative rotation errors RPE_r (bottom colour bar) of each method on *Ballroom* and *Museum*. The colour bar on the right shows the relative scaling of colour.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior
Publication Status	☑ Published □ Accepted for Publication □ Submitted for Publication □ Unpublished and unsubmitted work written in a manuscript style
Publication Details	**Bian, W**., Wang, Z., Li, K., Bian, J. W., & Prisacariu, V. A. (2023). Nope-nerf: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

Student Confirmation

Student Name:	Wenjing Bian		
Contribution to the Paper	• Conception and validation of the idea • Design of the architecture and framework • Implementation and running of all experiments • Research of prior work • Writing and presentation of the paper		
Signature	Wenjing Bian	Date	April 22, 2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Victor Adrian Prisacariu		
Supervisor comments The description is accurate		
Signature VPA	Date	April 22, 2025

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 6

PoRF: Pose Residual Field for Accurate Neural Surface Reconstruction

In the previous chapter, we focused on jointly optimising scene geometry and camera poses without any initial pose estimates. However, a common practical scenario involves having coarse pose initialisations from systems like COLMAP or ARKit, while still requiring further refinement to improve the reconstructed scene geometry. Although the problem setup appears similar, refining pre-initialised poses jointly with scene reconstruction remains a challenging task. In practice, gains in pose accuracy and reconstruction quality are often marginal. A primary challenge lies in the common practice of optimising pose parameters independently for each frame, without enforcing global consistency across the full trajectory. Additionally, the heavy reliance on RGB photometric loss introduces inherent ambiguities in the optimisation process, often resulting in convergence to local minima and producing noisy or distorted geometry.

In this chapter, we propose the Pose Residual Field (PoRF) for joint pose–scene

optimisation, adopting NeuS [64] as the neural surface representation. Given pose initialisations and frame indices as input, PoRF uses an MLP to predict pose residuals. As the MLP parameters are shared across all frames, the model implicitly captures global information across the full trajectory, leading to more robust and consistent pose optimisation. Additionally, we introduce image feature correspondence priors through an epipolar geometry loss to further regularise the optimisation, which altogether results in improved pose accuracy and reconstruction quality.

To summarise, our contributions are as follows:

- We propose a novel approach for pose refinement during neural surface reconstruction that leverages implicit global constraints across the trajectory.
- We introduce an epipolar geometry loss to incorporate correspondence priors into the optimisation process.
- We demonstrate that the proposed method improves pose accuracy and reconstruction quality through evaluation on several benchmarks.

The main content presented below in this chapter is based on our paper *PoRF: Pose Residual Field for Accurate Neural Surface Reconstruction*, published at the *International Conference on Learning Representations (ICLR) 2024*.

PoRF: Pose Residual Field for Accurate Neural Surface Reconstruction

Jia-Wang Bian [1, 2] Wenjing Bian[1] Victor Adrian Prisacariu[1] Philip Torr[2]
[1]Active Vision Lab [2]Torr Vision Group
University of Oxford

Abstract

*Neural surface reconstruction is sensitive to the camera pose noise, even if state-of-the-art pose estimators like COLMAP or ARKit are used. More importantly, existing Pose-NeRF joint optimisation methods have struggled to improve pose accuracy in challenging real-world scenarios. To overcome the challenges, we introduce the pose residual field (**PoRF**), a novel implicit representation that uses an MLP for regressing pose updates. This is more robust than the conventional pose parameter optimisation due to parameter sharing that leverages global information over the entire sequence. Furthermore, we propose an epipolar geometry loss to enhance the supervision that leverages the correspondences exported from COLMAP results without the extra computational overhead. Our method yields promising results. On the DTU dataset, we reduce the rotation error by 78% for COLMAP poses, leading to the decreased reconstruction Chamfer distance from 3.48mm to 0.85mm. On the MobileBrick dataset that contains casually captured unbounded 360-degree videos, our method refines ARKit poses and improves the reconstruction F1 score from 69.18 to 75.67, outperforming that with the dataset provided ground-truth pose (75.14). Moreover, we integrate our method into the Nerfstudio library, consistently improving performance in diverse challenging scenes. These achievements demonstrate the efficacy of our approach in refining camera poses in real-world scenarios.*

1. Introduction

Object reconstruction from multi-view images is a fundamental problem in computer vision. Recently, neural surface reconstruction (NSR) methods have significantly advanced in this field [1, 2]. These approaches draw inspiration from implicit scene representation and volume rendering techniques that were used in neural radiance fields (NeRF) [3]. In NSR, scene geometry is represented by using a signed distance function (SDF) field, learned by a multilayer perceptron (MLP) network trained with image-based rendering loss. Despite achieving high-quality reconstructions, these methods are sensitive to camera pose noise, a common issue in real-world applications, even when state-of-the-art pose estimation methods like COLMAP [4] or ARKit are used. An example of this sensitivity is evident in Fig. 1, where the reconstruction result with the COLMAP estimated poses shows poor quantitative accuracy and visible noise on the object's surface. In this paper, we focus on refining the inaccurate camera pose to improve neural surface reconstruction.

Recent research has explored the joint optimisation of camera pose and NeRF, while they are not designed for accurate neural surface reconstruction in real-world scenes. As shown in Fig. 1, existing methods such as BARF [5] and SPARF [6] are hard to improve reconstruction accuracy on the DTU dataset [7] via pose refinement. We regard the challenges as stemming from independent pose representation and weak supervision. Firstly, existing methods typically optimise pose parameters for each image independently. This approach disregards global information over the entire sequence and leads to poor accuracy. Secondly, the colour rendering loss employed in the joint optimisation process exhibits ambiguity, creating many false local minimums. To overcome these challenges, we introduce a novel implicit pose representation and a robust epipolar geometry loss into the joint optimisation framework.

The proposed implicit pose representation is named *Pose Residual Field* (PoRF), which employs an MLP network to learn the pose residuals. The MLP network takes the frame index and initial camera pose as inputs, as illustrated in Fig. 2. Notably, as the MLP parameters are shared across all frames, it is able to capture the underlying global information over the entire trajectory for boosting performance. As shown in Fig. 3, the proposed PoRF shows significantly improved accuracy and faster convergence than the conventional pose parameter optimisation approach.

Moreover, we use feature correspondences in the proposed epipolar geometry loss to enhance supervision. Although correspondences have been used in SPARF [6], it depends on expensive dense matching. As shown in Fig. 3, our method demonstrates similarly excellent performance with both sparse matches by handcraft methods (SIFT [8])

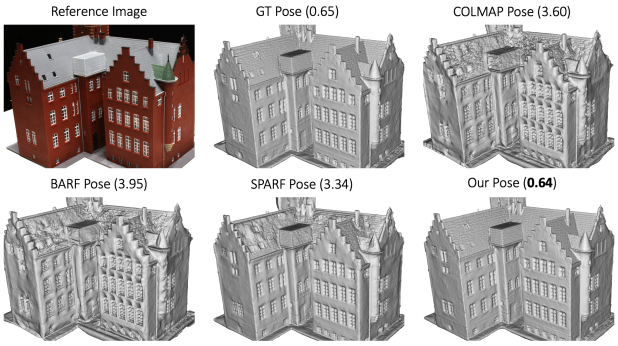


Figure 1. Reconstruction results on the DTU dataset (*scan24*). All meshes were generated by using Voxurf [2]. The Chamfer Distance (mm) is reported. BARF [5], SPARF [6], and our method all take the COLMAP pose as the initial pose. More results are illustrated in the supplementary material.

and dense matches by deep learning methods (LoFTR [9]). Besides, in contrast to SPARF which fuses correspondences and NeRF-rendered depths to compute reprojection loss, our approach uses the epipolar geometry loss that solely involves poses and correspondences, without relying on the rendered depths by NeRF that can be inaccurate and limit pose accuracy. Notably, as NeRF rendering is time-consuming, SPARF is constrained in the number of correspondences it can use in each iteration, while our method allows for the utilisation of an arbitrary number of matches.

We conduct a comprehensive evaluation on both DTU [7] and MobileBrick [10] datasets. Firstly, on the DTU dataset, our method takes the COLMAP pose as initialisation and reduces the rotation error by 78%. This decreases the reconstruction Chamfer Distance from 3.48mm to 0.85mm, where we use Voxurf [2] for reconstruction. Secondly, on the MobileBrick dataset, our method is initialised with ARKit poses and increases the reconstruction F1 score from 69.18 to 75.67. It slightly outperforms the provided imperfect GT pose in reconstruction (75.14) and achieves state-of-the-art performance. Moreover, we integrate our method into the Nerfstudio library [11], and the resulting method outperforms the original pose optimisation method in different challenging scenes. These results demonstrate the effectiveness of our approach in refining camera poses and improving reconstruction accuracy, especially in real-world scenarios where the initial poses might not be perfect.

In summary, our contributions are as follows:

1. We introduce a novel approach that optimises camera pose within neural surface reconstruction. It utilises the proposed *Pose Residual Field* (PoRF) and a robust epipolar geometry loss, enabling accurate and efficient refinement of camera poses.

2. The proposed method demonstrates its effectiveness in refining both COLMAP and ARKit poses, resulting in high-quality reconstructions on the DTU dataset and achieving state-of-the-art performance on the MobileBrick dataset.

2. Related Work

Neural surface reconstruction. Object reconstruction from multi-view images is a fundamental problem in computer vision. Traditional multi-view stereo (MVS) methods [12, 13] explicitly find dense correspondences across images to compute depth maps, which are fused together to obtain the final point cloud. The correspondence search and depth estimation processes are significantly boosted by the deep learning based approaches [14–16]. Recently, Neural Radiance Field (NeRF) [3] has been proposed to implicitly reconstruct the scene geometry, and it allows for extracting object surfaces from the implicit representation. To improve performance, VolSDF [17] and NeuS [1] proposed to use an implicit SDF field for scene representation. MonoSDF [18] and NeurIS [19] proposed to leverage the external monocular geometry prior. HF-NeuS [20] introduced an extra displacement network for learning surface details. NeuralWarp [21] used patch warping to guide surface optimisation. Geo-NeuS [22] and RegSDF [23] proposed to leverage the sparse points generated by SfM. Neuralangelo [24] proposed a coarse-to-fine method on the hash grids. Voxurf [2] proposed a voxel-based representation that achieves high accuracy with efficient training.

Joint NeRF and pose optimisation. NeRFmm [25] demonstrates the possibility of jointly learning or refining camera parameters alongside the NeRF framework [3]. BARF [5] introduces a solution involving coarse-to-fine positional encoding to enhance the robustness of joint optimisation. In the case of SC-NeRF [26], both intrinsic and extrinsic camera parameters are proposed for refinement. In separate approaches, SiNeRF [27] and GARF [28] suggest employing distinct activation functions within NeRF networks to facilitate pose optimiation. Nope-NeRF [29] employs an external monocular depth estimation model to assist in refining camera poses. L2G [30] puts forth a local-to-global scheme, wherein the image pose is computed from learned multiple local poses. Notably similar to our approach is SPARF [6], which also incorporates correspondences and evaluates its performance on the challenging DTU dataset [7]. However, it is tailored to sparse-view scenarios and lacks the accuracy achieved by our method when an adequate number of images is available.

3. Preliminary

Volume rendering with SDF representation We adhere to the NeuS [1] methodology to optimise the neural surface

reconstruction from multi-view images. The approach involves representing the scene using an implicit SDF field, which is parameterised by an MLP. To render an image pixel, a ray originates from the camera centre o and extends through the pixel along the viewing direction v, described as $\{p(t) = o + tv | t \geq 0\}$. By employing volume rendering [31], the colour for the image pixel is computed by integrating along the ray using N discrete sampled points $\{p_i = o + t_i v | i = 1, ..., N, t_i < t_{i+1}\}$:

$$\hat{C}(r) = \sum_{i=1}^{N} T_i \alpha_i c_i, \;\; T_i = \prod_{j=1}^{i-1} (1 - \alpha_j), \tag{1}$$

In this equation, α_i represents the opacity value, and T_i denotes the accumulated transmittance. The primary difference between NeuS and NeRF [3] lies in how α_i is formulated. In NeuS, α_i is computed using the following expression:

$$\alpha_i = \max\left(\frac{\Phi_s(f(p(t_i))) - \Phi_s(f(p(t_{i+1})))}{\Phi_s(f(p(t_i)))}, 0 \right), \tag{2}$$

In this context, $f(x)$ is the SDF function, and $\Phi_s(x) = (1 + e^{-sx})^{-1}$ denotes the Sigmoid function, where the parameter s is automatically learned during the training process.

Neural surface reconstruction loss. In line with NeuS [1], our objective is to minimise the discrepancy between the rendered colours and the ground truth colours. To achieve this, we randomly select a batch of pixels and their corresponding rays in world space $P = \{C_k, o_k, v_k\}$ from an image in each iteration. Here, C_k represents the pixel colour, o_k is the camera centre, and v_k denotes the viewing direction. We set the point sampling size as n and the batch size as m.

The neural surface reconstruction loss function is defined as follows:

$$\mathcal{L}_{NSR} = \mathcal{L}_{colour} + \lambda \mathcal{L}_{reg}. \tag{3}$$

The colour loss $\mathcal{L}_{colour}$ is calculated as:

$$\mathcal{L}_{colour} = \frac{1}{m} \sum_{k} \left\| \hat{C}_k - C_k \right\|_1. \tag{4}$$

The term $\mathcal{L}_{reg}$ incorporates the Eikonal term [32] applied to the sampled points to regularise the learned SDF, which can be expressed as:

$$\mathcal{L}_{reg} = \frac{1}{nm} \sum_{k,i} (\|\nabla f(\hat{\mathbf{p}}_{k,i})\|_2 - 1)^2. \tag{5}$$

Here, f is the learned SDF, and $\hat{\mathbf{p}}_{k,i}$ represents the sampled points for the k-th pixel in the i-th ray. λ controls the influence of the regularisation term in the overall loss function.

Joint pose optimisation with NSR. Inspired by the previous work [5, 25] that jointly optimises pose parameters with NeRF [3], we propose a naive joint pose optimisation with neural surface reconstruction and use Eqn. 3 as the loss function. This is denoted as the baseline method in Fig. 3, and in the next section, we present the proposed method for improving performance.

4. Method

Pose residual field (PoRF). We approach camera pose refinement as a regression problem using an MLP. In Fig. 2, we illustrate the process where the input to the MLP network is a 7D vector, consisting of a 1D image index (i) and a 6D initial pose (r, t). The output of the MLP is a 6D pose residual $(\Delta r, \Delta t)$. This residual is combined with the initial pose to obtain the final refined pose (r', t'). In this context, rotation is represented by 3D axis angles.

To encourage small pose residual outputs at the beginning of the optimisation, we multiply the MLP output by a fixed small factor (α). This ensures an effective initialisation (the refined pose is close to the initial pose), and the correct pose residuals can be gradually learned with optimisation. Formally, the composition of the initial pose and the learned pose residual is expressed as follows:

$$\begin{cases} \hat{r} = r + \alpha \cdot \Delta r \\ \hat{t} = t + \alpha \cdot \Delta t \end{cases} \tag{6}$$

The resulting final pose $(\hat{r}, \hat{t})$ is utilised to compute the loss, and the gradients can be backpropagated to the MLP parameters during the optimisation process. In practice, we use a shallow MLP with 2 hidden layers, which we find is sufficient for object reconstruction.

Intuition on the PoRF. Our method draws inspiration from the coordinates-MLP. Similar to NeRF [3], in which the position and view direction of a point are taken as inputs by the MLP to learn its density and colour, our method takes the initial position (translation) and view direction (rotation) of the camera as inputs to learn the camera's pose residual. Also, our approach incorporates the frame index as an additional input, serving as a temporal coordinate.

The MLP parameters are shared across all frames, which enables our method to extract global information from the entire sequence. The inclusion of the frame index allows our method to learn local information for each frame. Overall, the utilisation of both global and local information empowers our method to effectively handle various challenges and learn accurate camera poses.

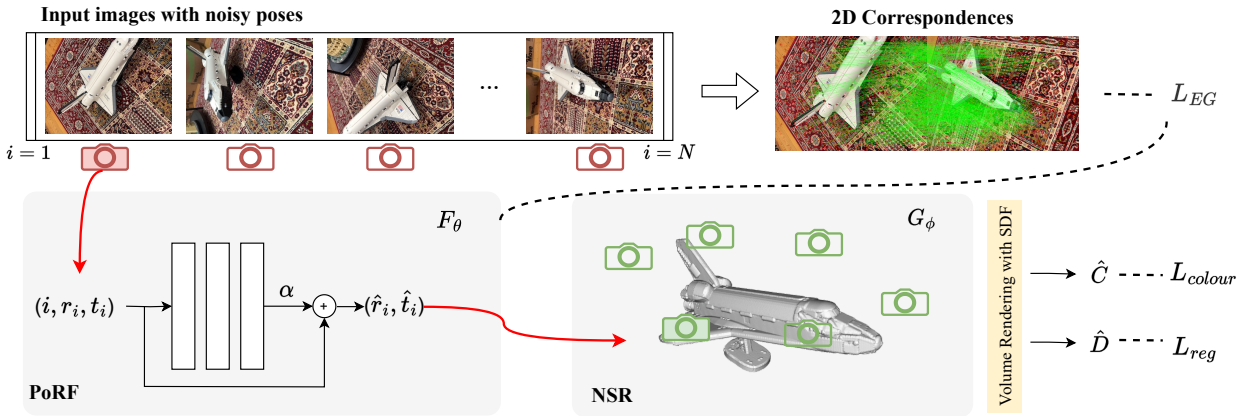


Figure 2. Joint optimisation pipeline. The proposed model consists of a pose residual field F_θ and a neural surface reconstruction module G_ϕ. PoRF takes the frame index and the initial camera pose as input and employs an MLP to learn the pose residual, which is composited with the initial pose to obtain the predicted pose. The output pose is used to compute the neural rendering losses with the NSR module and the epipolar geometry loss with pre-computed 2D correspondences. Parameters θ and ϕ are updated during back-propagation.

The parameter sharing in our method leads to better robustness than the conventional pose refinement method. Specifically, the previous method directly back-propagates gradients to per-frame pose parameters, so the noisy supervision and inappropriate learning rates can make certain pose parameters trapped in false local minima, causing the optimisation to diverge. In contrast, our method can leverage information from other frames to prevent certain frames from being trapped in false local minima. A quantitative comparison is illustrated in Fig. 3, where the performance of L2 (utilising PoRF) notably surpasses that of L1 (excluding PoRF).

Epipolar geometry loss. To improve the supervision, we introduce an epipolar geometry loss that makes use of feature correspondences. The correspondences can be either sparse or dense, and they can be obtained through traditional handcrafted descriptors, like SIFT [8], or deep learning techniques, like LoFTR [9]. As a default option, we utilise SIFT matches, exported from COLMAP pose estimation results, to avoid introducing extra computational cost.

In each training iteration, we randomly sample n matched image pairs and calculate the Sampson distance (Eqn. 8) for the correspondences. By applying a threshold δ, we can filter out outliers and obtain m inlier correspondences. We represent the inlier rate for each pair as p_i, and the Sampson error for each inlier match as e_k. The proposed epipolar geometry loss is then defined as:

$$\mathcal{L}_{EG} = \frac{1}{n} \sum_i w_i \left(\frac{1}{m} \sum_k e_k \right). \tag{7}$$

Here, $w_i = {p_i}^2$ is the loss weight that mitigates the influence of poorly matched pairs. The computation of the Sampson distance between two points, denoted as x and x', with the fundamental matrix F obtained from the (learned) camera poses and known intrinsics, is expressed as:

$$d_{\text{Sampson}}(x, x', F) = \frac{(x'^T F x)^2}{(Fx)_1^2 + (Fx)_2^2 + (F^T x')_1^2 + (F^T x')_2^2}. \tag{8}$$

Here, subscripts 1 and 2 refer to the first and second components of the respective vectors.

Training objectives. The overall loss function in our proposed joint optimisation method combines both Eqn. 3 and Eqn. 7, as follows:

$$\mathcal{L} = \mathcal{L}_{NSR} + \beta \mathcal{L}_{EG}, \tag{9}$$

where β is a hyperparameter that controls loss weights. The first term $\mathcal{L}_{NSR}$ represents the loss related to neural surface reconstruction, which backpropagates the gradients to both PoRF and rendering network parameters. The second term $\mathcal{L}_{EG}$ represents the loss associated with the epipolar geometry loss, which backpropagates the gradients to PoRF only.

5. Experiments

Experiment setup. We perform our experiments on both the DTU [7] and MobileBrick [10] datasets. Following previous methods such as [2] and [10], our evaluation uses the provided 15 test scenes from DTU and 18 test scenes from MobileBrick. We assess both the accuracy of camera poses and the quality of surface reconstructions. The baseline

Table 1. Absolute pose accuracy evaluation on the DTU dataset. For all methods, the COLMAP pose is used as the initial pose. The upper section presents rotation errors in degrees, while the lower section displays translation errors in millimetres (mm).

Scan	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	mean
COLMAP (Initial)	0.68	0.67	0.63	0.62	0.75	0.67	0.64	0.68	0.56	0.66	0.62	0.69	0.66	0.64	0.63	0.65
BARF [5]	0.67	0.73	0.67	0.79	0.67	0.83	0.47	0.59	0.55	0.62	0.75	0.67	0.67	0.70	0.84	0.69
L2G [30]	0.67	0.65	0.69	32.04	0.80	0.68	0.49	0.66	0.76	0.61	29.02	6.41	0.69	2.63	2.47	5.29
SPARF [6]	0.40	0.62	0.43	0.40	0.72	0.39	0.30	0.31	0.48	0.33	0.36	0.39	0.19	0.24	0.36	0.39
Ours	**0.09**	**0.13**	**0.10**	**0.13**	**0.13**	**0.16**	**0.13**	**0.12**	**0.26**	**0.12**	**0.12**	**0.16**	**0.15**	**0.11**	**0.13**	**0.14**
COLMAP (Initial)	0.79	**1.06**	0.81	0.81	0.80	0.89	0.88	**1.07**	2.07	1.07	1.05	1.22	1.03	**0.95**	**1.02**	1.03
BARF [5]	1.53	2.19	1.80	2.50	2.32	2.31	2.06	2.65	3.45	2.27	2.29	2.59	3.29	3.15	3.13	2.50
L2G [30]	1.50	1.90	1.90	115	2.07	2.45	1.89	2.77	5.04	1.99	172	88.6	2.94	12.3	21.1	28.9
SPARF [6]	3.66	4.76	3.89	2.22	5.68	2.57	2.32	2.96	4.88	3.21	2.94	2.85	1.96	2.23	2.65	3.25
Ours	**0.76**	1.11	**0.77**	**0.80**	**0.79**	**0.81**	**0.84**	1.08	**2.05**	**1.02**	**1.03**	**1.15**	**0.99**	0.97	1.07	**1.02**

Table 2. Reconstruction accuracy evaluation on the DTU dataset. The evaluation metric is Chamfer distance, expressed in millimetres (mm).

Method	Pose	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	mean
NeRF [3]	GT	1.83	2.39	1.79	0.66	1.79	1.44	1.50	1.20	1.96	1.27	1.44	2.61	1.04	1.13	0.99	1.54
IDR [33]		1.63	1.87	0.63	0.48	1.04	0.79	0.77	1.33	1.16	0.76	0.67	0.90	0.42	0.51	0.53	0.90
VolSDF [17]		1.14	1.26	0.81	0.49	1.25	0.70	0.72	1.29	1.18	0.70	0.66	1.08	0.42	0.61	0.55	0.86
PointNeRF [34]		0.87	2.06	1.20	1.01	1.01	1.39	0.80	**1.04**	**0.92**	0.74	0.97	**0.76**	0.56	0.90	1.05	1.02
NeuS [1]		0.83	0.98	0.56	0.37	1.13	**0.59**	**0.60**	1.45	0.95	0.78	**0.52**	1.43	**0.36**	**0.45**	**0.45**	0.77
Voxurf [2]		**0.65**	**0.74**	**0.39**	**0.35**	**0.96**	0.64	0.85	1.58	1.01	**0.68**	0.60	1.11	0.37	**0.45**	0.47	**0.72**
Voxurf	COLMAP	3.60	4.36	3.62	3.00	4.44	3.52	2.64	4.11	2.85	3.86	2.98	3.86	2.76	3.26	3.34	3.48
	BARF	3.95	4.35	3.84	3.02	4.38	4.09	2.02	3.68	2.56	3.62	3.61	4.06	2.63	3.04	4.14	3.53
	SPARF	3.34	5.30	1.35	2.75	6.17	2.81	1.85	3.24	1.85	2.17	2.05	5.30	1.03	1.62	2.45	2.89
	Ours	**0.64**	**0.84**	**0.59**	**0.48**	**1.10**	**1.02**	**0.90**	**1.43**	**1.10**	**0.83**	**0.86**	**1.11**	**0.58**	**0.65**	**0.67**	**0.85**

methods for comparison includes BARF [5], L2G [30], and SPARF [6]. For a fair comparison, all methods, including ours, take the same initial pose inputs—namely, COLMAP poses on the DTU dataset and ARKit poses on the MobileBrick dataset. Here, we consider all methods to be operating in a two-stage manner, so only the pose refinement is conducted by these methods, and we utilise Voxurf [2] with the refined poses for the reconstruction step for all methods.

Implementation details. The hyperparameters used for training the surface reconstruction align with those utilised in NeuS [1]. To calculate the proposed epipolar geometry loss, we randomly sample 20 image pairs in each iteration and distinguish inliers and outliers by using a threshold of 20 pixels. The entire framework undergoes training for 50,000 iterations, which takes 2.5 hours in one NVIDIA A40 GPU. However, it's noteworthy that convergence is achieved rapidly, typically within the first 5,000 iterations. More details are in the supplementary materials.

5.1. Comparisons

Results on DTU. The pose refinement results are presented in Table 1. It shows that BARF [5] and L2G [30] are unable to improve the COLMAP pose on DTU, and L2G diverges in several scenes. While SPARF [6] shows a slight improvement in rotation, it remains limited, and it results in worse translation. Besides, we also evaluated Nope-NeRF [29], but it diverged on DTU and MobileBrick scenes due to the bad initialisation of the depth scale. Overall, our proposed method achieves a substantial reduction of 78% in rotation error and shows a minor improvement in translation error.

The reconstruction results are presented in Tab. 2, where we run Voxurf [2] for reconstruction. It shows that the reconstruction with the COLMAP pose falls behind previous methods that use the GT pose. BARF [5] is hard to improve performance. While SPARF [6] can slightly improve performance, it is limited. When employing the refined pose by our method, the reconstruction accuracy becomes comparable to most of the previous methods that use the GT pose, although there is still a minor gap between our method and Voxurf with the GT pose, *i.e.*, 0.85mm vs. 0.72mm.

Results on MobileBrick. The pose results are summarised in Tab. 3. We initialise all baseline methods, including ours, with the ARKit pose. Note that the provided GT pose in this dataset is imperfect, which is obtained by pose refinement from human labelling. The results show that our method is closer to the GT pose than others. More importantly, our method can consistently improve pose accuracy in all scenes, while other methods cannot.

The reconstruction results are presented in Tab. 4. It shows that the refined pose by all methods can lead to an overall improvement in reconstruction performance. Compared with BARF [5] and SPARF [6], our method shows more significant improvement. As a result, our approach

Table 3. Absolute pose accuracy on the MobileBrick dataset. The ARKit pose serves as the initial pose for all methods. Note that the provided GT pose in this dataset is imperfect.

Metrics	Rotation Error (Deg.)					Translation Error (mm)				
Scene	ARKit	BARF	L2G	SPARF	Ours	ARKit	BARF	L2G	SPARF	Ours
aston	0.43	0.22	0.22	0.22	**0.14**	2.49	1.21	1.21	1.01	**0.70**
audi	0.64	0.46	0.52	1.10	**0.28**	3.38	2.25	2.24	4.35	**1.11**
beetles	0.49	0.52	0.57	0.17	**0.14**	1.40	1.29	2.06	**1.21**	1.23
big_ben	0.44	0.33	0.49	0.27	**0.26**	4.23	3.03	2.13	2.38	**1.82**
boat	0.61	0.24	0.38	0.26	**0.22**	1.51	0.97	1.29	1.12	**0.98**
bridge	0.44	0.22	0.18	0.32	**0.22**	1.08	1.29	1.12	1.26	**0.78**
cabin	0.61	0.34	0.32	0.30	**0.26**	1.34	1.44	1.55	1.09	**0.93**
camera	0.30	0.23	0.26	**0.17**	0.18	0.86	1.06	1.07	0.67	**0.61**
castle	0.55	4.77	4.79	4.77	**0.30**	2.42	17.1	21.1	24.12	**2.55**
colosseum	0.46	0.49	0.35	0.95	**0.28**	3.67	4.21	4.41	11.26	**3.25**
convertible	0.40	0.27	0.25	0.23	**0.13**	1.65	1.18	1.21	1.31	**0.92**
ferrari	0.43	0.20	0.20	0.23	**0.13**	1.89	1.10	1.19	0.80	**0.59**
jeep	0.34	0.24	0.22	0.28	**0.23**	0.98	1.13	1.08	0.65	**0.50**
london_bus	0.72	0.70	0.73	0.50	**0.42**	3.62	2.97	2.98	3.61	**2.63**
motorcycle	0.34	0.26	0.29	0.27	**0.20**	0.97	1.01	1.10	0.82	**0.60**
porsche	0.28	0.19	0.20	0.20	**0.15**	1.29	1.16	1.18	0.88	**0.71**
satellite	0.41	0.23	0.26	**0.15**	0.17	1.45	0.92	1.11	**0.97**	0.98
space_shuttle	0.31	0.28	0.26	0.58	**0.18**	1.45	1.81	1.76	3.43	**1.30**
mean	0.46	0.57	0.58	0.61	**0.22**	1.90	2.50	2.84	3.39	**1.23**

Table 4. Reconstruction accuracy evaluation on the MobileBrick dataset. Note that the provided GT pose in this dataset is imperfect. The results are averaged over all 18 test scenes.

Methods	Pose	$\sigma = 2.5mm$			$\sigma = 5mm$			Chamfer ↓
		Accu. (%) ↑	Rec. (%) ↑	F1 ↑	Accu. (%) ↑	Rec. (%) ↑	F1 ↑	(mm)
TSDF-Fusion [35]	GT	42.07	22.21	28.77	73.46	42.75	53.39	13.78
BNV-Fusion [36]		41.77	25.96	33.27	71.20	47.09	55.11	9.60
Neural-RGBD [37]		20.61	10.66	13.67	39.62	22.06	27.66	22.78
COLMAP [4]		74.89	68.20	71.08	93.79	84.53	88.71	5.26
NeRF [3]		47.11	40.86	43.55	78.07	69.93	73.45	7.98
NeuS [1]		77.35	70.85	73.74	93.33	86.11	89.30	4.74
Voxurf [2]		78.44	72.41	75.13	93.95	87.53	90.41	4.71
Voxurf	ARKit	71.90	66.98	69.18	91.81	85.71	88.43	5.30
	BARF	77.12	71.36	73.96	93.47	87.56	90.22	4.83
	SPARF	76.10	70.44	72.99	92.73	86.66	89.39	4.95
	Ours	**79.01**	**72.94**	**75.67**	**94.07**	**87.84**	**90.65**	**4.67**

surpasses the provided GT pose in terms of reconstruction F1 score, *i.e.*, 75.67 vs 75.13, achieving state-of-the-art performance on the MobileBrick dataset.

5.2. Generalisation

The proposed MLP-based pose optimisation method can be readily plugged into different NeRF-like systems to improve performance. To demonstrate the universal use of our method, we integrate it into the Nerfstudio library [11], which is a modular framework for neural radiance field development. We compare our method with the recommended baseline method (Nerfacto), which integrates multiple methods into one, including pose optimisation. For a fair comparison, we replace the original pose parameter optimisation module with our proposed PoRF MLP, and we do not use the proposed loss function. Tab. 5 provides the evaluation results on the Nerfstudio dataset. The dataset comprises ten in-the-wild, 360-degree captures obtained using either a mobile phone or a mirror-less camera with a fish-eye lens. The data was processed using COLMAP or the Polycam app to obtain camera poses and intrinsic parameters. It shows that our method can consistently boost novel view rendering performance in diverse scenes.

5.3. Ablation studies

We conduct extensive ablation studies on the DTU dataset [7]. The pose errors during training are shown in Fig. 3, where we compare six different settings and we can draw some key observations from the ablation studies:

PoRF vs pose parameter optimisation: L2 surpasses L1 in terms of accuracy, thereby highlighting the benefits of utilising the Pose Residual Field (PoRF) in contrast to conventional pose parameter optimisation within the same

Table 5. Novel view rendering results on the Nerfstudio dataset. We replace the pose optimisation module in the Nerfacto with our PoRF MLP, leading to consistently improved novel view rendering accuracy. Note that we do not use the proposed loss function here.

Scenes	Ours			Nerfacto		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR	SSIM	LPIPS
Egypt	**23.08**	**0.743**	**0.245**	20.75	0.623	0.305
person	**25.87**	**0.742**	**0.261**	24.33	0.643	0.299
kitchen	**20.72**	**0.796**	**0.271**	19.60	0.73	0.298
plane	**22.44**	**0.672**	**0.356**	19.72	0.557	0.440
dozer	**21.16**	**0.760**	**0.266**	20.26	0.716	0.279
floating tree	**20.82**	**0.769**	**0.204**	20.67	0.733	0.211
aspen	**17.16**	**0.344**	**0.611**	16.48	0.292	0.657
stump	**22.62**	**0.820**	**0.194**	21.96	0.753	0.213
sculpture	**22.28**	**0.692**	**0.296**	21.74	0.652	0.315
Giannini Hall	**19.29**	**0.590**	**0.406**	17.66	0.491	0.444
mean	**21.54**	**0.693**	**0.311**	20.32	0.619	0.346

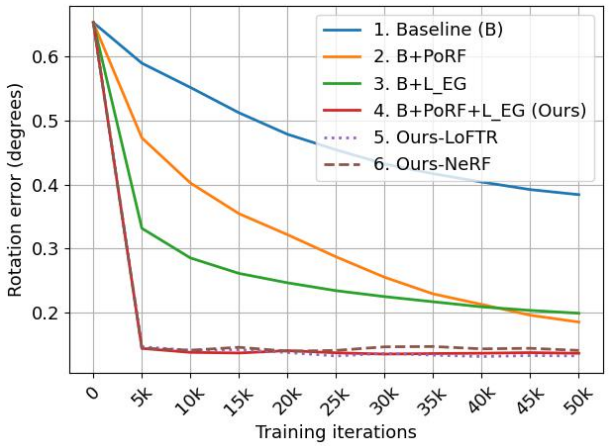


Figure 3. Pose errors during training on the DTU dataset. The results are averaged over 15 test scenes. Baseline (B) denotes the naive joint optimisation of NSR and pose parameters.

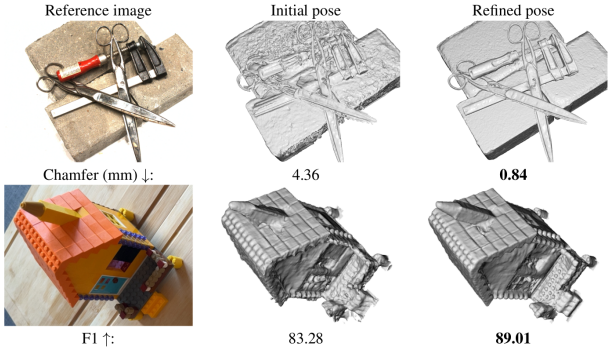


Figure 4. Reconstruction results on DTU (top) and MobileBrick (bottom) datasets. The initial pose denotes the COLMAP pose on DTU, and the ARKit pose on MobileBrick. We use the standard evaluation metrics, *i.e.*, Chamfer distance (mm) for DTU and F1 score for MobileBrick.

baseline framework, where solely the neural rendering loss is employed. Upon the integration of the proposed epipolar geometry loss L_{EG} with the baseline approach, L4 outperforms L3, substantiating the consistency of the observed outcomes.

Efficacy of the epipolar geometry loss: L3 demonstrates faster convergence when compared to L1, offering substantiation of the efficacy of the proposed L_{EG} loss within the context of pose parameter optimisation. Upon incorporating the proposed Pose Residual Field (PoRF), L4 surpasses L2, thus reinforcing the same conclusion.

Ablation over correspondences: we compare handcrafted matching methods (L4 representing SIFT [8]) with deep learning-based methods (L5 denoting LoFTR [9]). On average, SIFT yields 488 matches per image pair, while LoFTR generates a notably higher count of 5394 matches. The results indicate that both methods achieve remarkably similar levels of accuracy and convergence. It's worth noting that the SIFT matches are derived from COLMAP results, so their utilisation does not entail additional computational overhead.

Ablation over scene representations: We substitute the SDF-based representation (L4) with the NeRF representation [3], denoted as L6. This adaptation yields comparable performance, highlighting the versatility of our method across various implicit representations.

6. Conclusion

This paper introduces the concept of the Pose Residual Field (**PoRF**) and integrates it with an epipolar geometry loss to facilitate the joint optimisation of camera pose and neural surface reconstruction. Our approach leverages the initial camera pose estimation from COLMAP or

ARKit and employs these newly proposed components to achieve high accuracy and rapid convergence. By incorporating our refined pose in conjunction with Voxurf, a cutting-edge surface reconstruction technique, we achieve state-of-the-art reconstruction performance on the MobileBrick dataset. Furthermore, when evaluated on the DTU dataset, our method exhibits comparable accuracy to previous approaches that rely on ground-truth pose information.

Acknowledgement

The authors gratefully acknowledge the financial support provided by Apple. This work is also supported by the UKRI grant: Turing AI Fellowship EP/W002981/1 and EPSRC/MURI grant: EP/N019474/1. We also thank the Royal Academy of Engineering and FiveAI.

References

[1] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," in *NeurIPS*, 2021. 1, 2, 3, 5, 6, 10, 11, 13

[2] T. Wu, J. Wang, X. Pan, X. Xu, C. Theobalt, Z. Liu, and D. Lin, "Voxurf: Voxel-based efficient and accurate neural surface reconstruction," in *ICLR*, 2023. 1, 2, 4, 5, 6, 10, 11, 12, 15, 16

[3] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, pp. 405–421, Springer, 2020. 1, 2, 3, 5, 6, 7, 10

[4] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," *CVPR*, 2016. 1, 6, 11

[5] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in *ICCV*, pp. 5741–5751, 2021. 1, 2, 3, 5, 11, 12

[6] P. Truong, M.-J. Rakotosaona, F. Manhardt, and F. Tombari, "Sparf: Neural radiance fields from sparse and noisy poses," in *CVPR*, pp. 4190–4200, 2023. 1, 2, 5, 11, 12, 13

[7] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs, "Large scale multi-view stereopsis evaluation," in *CVPR*, pp. 406–413, 2014. 1, 2, 4, 6, 10, 12

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, pp. 91–110, 2004. 1, 4, 7, 10

[9] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "LoFTR: Detector-free local feature matching with transformers," *CVPR*, 2021. 2, 4, 7, 10

[10] K. Li, J.-W. Bian, R. Castle, P. H. Torr, and V. A. Prisacariu, "Mobilebrick: Building lego for 3d reconstruction on mobile devices," in *CVPR*, pp. 4892–4901, 2023. 2, 4, 10

[11] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, *et al.*, "Nerfstudio: A modular framework for neural radiance field development," in *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–12, 2023. 2, 6

[12] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *ECCV*, 2016. 2

[13] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Reconstructing building interiors from images," in *ICCV*, pp. 80–87, IEEE, 2009. 2

[14] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," in *ECCV*, pp. 767–783, 2018. 2

[15] Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang, and L. Quan, "Recurrent mvsnet for high-resolution multi-view stereo depth inference," in *CVPR*, pp. 5525–5534, 2019. 2

[16] J. Zhang, Y. Yao, S. Li, Z. Luo, and T. Fang, "Visibility-aware multi-view stereo network," *BMVC*, 2020. 2

[17] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 2, 5

[18] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, and A. Geiger, "Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction," *NeurIPS*, 2022. 2

[19] J. Wang, P. Wang, X. Long, C. Theobalt, T. Komura, L. Liu, and W. Wang, "Neuris: Neural reconstruction of indoor scenes using normal priors," in *ECCV*, pp. 139–155, Springer, 2022. 2

[20] Y. Wang, I. Skorokhodov, and P. Wonka, "Hf-neus: Improved surface reconstruction using high-frequency details," *NeurIPS*, vol. 35, pp. 1966–1978, 2022. 2

[21] F. Darmon, B. Bascle, J.-C. Devaux, P. Monasse, and M. Aubry, "Improving neural implicit surfaces geometry with patch warping," in *CVPR*, pp. 6260–6269, 2022. 2

[22] Q. Fu, Q. Xu, Y. S. Ong, and W. Tao, "Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction," *NeurIPS*, vol. 35, pp. 3403–3416, 2022. 2

[23] J. Zhang, Y. Yao, S. Li, T. Fang, D. McKinnon, Y. Tsin, and L. Quan, "Critical regularizations for neural surface reconstruction in the wild," in *CVPR*, pp. 6270–6279, 2022. 2

[24] Z. Li, T. Müller, A. Evans, R. H. Taylor, M. Unberath, M.-Y. Liu, and C.-H. Lin, "Neuralangelo: High-fidelity neural surface reconstruction," in *CVPR*, 2023. 2

[25] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu, "NeRF−−: Neural radiance fields without known camera parameters," *arXiv preprint arXiv:2102.07064*, 2021. 2, 3

[26] Y. Jeong, S. Ahn, C. Choy, A. Anandkumar, M. Cho, and J. Park, "Self-calibrating neural radiance fields," in *ICCV*, pp. 5846–5854, 2021. 2

[27] Y. Xia, H. Tang, R. Timofte, and L. Van Gool, "Sinerf: Sinusoidal neural radiance fields for joint pose estimation and scene reconstruction," *arXiv preprint arXiv:2210.04553*, 2022. 2

[28] S.-F. Chng, S. Ramasinghe, J. Sherrah, and S. Lucey, "Garf: gaussian activated radiance fields for high fidelity reconstruction and pose estimation," *arXiv e-prints*, pp. arXiv–2204, 2022. 2

[29] W. Bian, Z. Wang, K. Li, J.-W. Bian, and V. A. Prisacariu, "Nope-nerf: Optimising neural radiance field with no pose prior," in *CVPR*, pp. 4160–4169, 2023. 2, 5

[30] Y. Chen, X. Chen, X. Wang, Q. Zhang, Y. Guo, Y. Shan, and F. Wang, "Local-to-global registration for bundle-adjusting neural radiance fields," in *CVPR*, pp. 8264–8273, 2023. 2, 5, 11

[31] N. Max, "Optical models for direct volume rendering," *IEEE TVCG*, 1995. 3

[32] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, "Implicit geometric regularization for learning shapes," *arXiv preprint arXiv:2002.10099*, 2020. 3

[33] L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, B. Ronen, and Y. Lipman, "Multiview neural surface reconstruction by disentangling geometry and appearance," *NeurIPS*, vol. 33, 2020. 5, 10

[34] Q. Xu, Z. Xu, J. Philip, S. Bi, Z. Shu, K. Sunkavalli, and U. Neumann, "Point-nerf: Point-based neural radiance fields," in *CVPR*, pp. 5438–5448, 2022. 5

[35] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018. 6

[36] K. Li, Y. Tang, V. A. Prisacariu, and P. H. Torr, "Bnv-fusion: Dense 3d reconstruction using bi-level neural volume fusion," in *CVPR*, 2022. 6

[37] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies, "Neural rgb-d surface reconstruction," in *CVPR*, pp. 6290–6301, June 2022. 6

[38] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "Nerf++: Analyzing and improving neural radiance fields," *arXiv preprint arXiv:2010.07492*, 2020. 10

[39] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," *CVPR*, 2022. 10

[40] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, July 2022. 13

A. Experimental results

A.1. Datasets

The DTU [7] dataset contains a variety of object scans with 49 or 64 posed multi-view images for each scan. It covers different materials, geometry, and texture. We evaluate our method on DTU with the same 15 test scenes following previous work, including IDR [33], NeuS [1], and Voxurf [2]. The results are quantitatively compared by using Chamfer Distance, given the ground truth point clouds.

The MobileBrick [10] contains iPhone-captured 360-degree videos of different objects that are built with LEGO bricks. Therefore, the accurate 3D mesh, obtained from the LEGO website, is provided as the ground truth for evaluation. The ARKit camera poses and the manually refined ground-truth poses by the authors are provided, however, the accuracy is not comparable to the poses obtained by camera calibration such as that in the DTU dataset. We compare our method with other approaches on 18 test scenes by following the provided standard pipeline.

A.2. Evaluation metrics

To assess the accuracy of camera poses, we employ a two-step procedure. First, we perform a 7-degree-of-freedom (7-DoF) pose alignment to align the refined poses with the ground-truth poses. This alignment process ensures that the calculated poses are in congruence with the ground truth. Subsequently, we gauge the alignment's effectiveness by utilising the standard absolute trajectory error (ATE) metric, which quantifies the discrepancies in both rotation and translation between the aligned camera poses and the ground truth. Moreover, we utilise these aligned camera poses to train Voxurf [2] for object reconstruction. This ensures that the reconstructed objects are accurately aligned with the ground-truth 3D model, facilitating a meaningful comparison between the reconstructed and actual objects.

A.3. Implementation details

Coordinates normalisation for unbounded 360-degree scenes. In the DTU [7] and MobileBrick [10] datasets, images include distant backgrounds. The original NeuS [1] either uses segmentation masks to remove the backgrounds or reconstructs them using an additional NeRF model [3], following the NeRF++ method [38].

In contrast, we adopt the MipNeRF-360 approach [39], which employs a normalisation to handle the point coordinates efficiently. The normalisation allows all points in the scene to be modelled using a single MLP network. The normalisation function is defined as:

$$\text{contract}(\mathbf{x}) = \begin{cases} \mathbf{x} & \|\mathbf{x}\| \leq 1 \\ \left(2 - \frac{1}{\|\mathbf{x}\|}\right)\left(\frac{\mathbf{x}}{\|\mathbf{x}\|}\right) & \|\mathbf{x}\| > 1 \end{cases} \tag{10}$$

By applying this normalisation function, all points in the scene are effectively mapped to the range (0-2) in space. This normalised representation is used to construct an SDF field, which is then utilised for rendering images. Therefore, the proposed method can process images captured from 360-degree scenes without the need for segmentation masks.

Rendering network architectures. Our neural surface reconstruction code is built upon NeuS [1], which means that the network architectures for both SDF representation and volume rendering are identical. To provide more detail, the SDF network comprises 8 hidden layers, with each layer housing 256 nodes, and it employs the ELU activation function for nonlinearity. Similarly, the rendering network consists of 4 layers and utilises the same nonlinear activation functions. This consistency in architecture ensures that the SDF and rendering processes are aligned within our implementation.

PoRF MLP architectures. We employ a simple neural network structure for our PoRF (Pose Refinement) module, consisting of a 2-layer shallow MLP. Each layer in this MLP comprises 256 nodes, and the activation function employed is the Exponential Linear Unit (ELU) for introducing nonlinearity. The input to the PoRF MLP consists of a 7-dimensional vector. This vector is comprised of a 1-dimensional normalised frame index and a 6-dimensional set of pose parameters. The PoRF MLP's output is a 6-dimensional vector, which represents the pose residual.

Correspondence generation. By default, we export correspondences from COLMAP, which are generated using the SIFT descriptor [8] and the outliers are removed by using a RANSAC-based two-view geometry verification approach. These correspondences, while reliable, tend to be sparse and may contain noise. However, our method is designed to automatically handle outliers in such data. Additionally, we conduct experiments with correspondences generated by LoFTR [9]. These correspondences are dense, providing a denser point cloud representation. We adhere to the original implementation's guidelines to select high-confidence correspondences and further employ a RANSAC-based technique to eliminate outliers. Consequently, the final set of correspondences obtained through LoFTR remains denser than SIFT matches.

Training details. Our approach follows a two-stage pipeline. In the first stage, we undertake the training of the complete framework, conducting a total of 50,000 iterations using all available images. This comprehensive training enables the refinement of pose information for each image. In the second stage, we employ Voxurf [2] for the purpose of reconstruction. It's important to note that this reconstruction is executed solely using the training images. This step ensures a fair comparison with prior methods, encompassing both surface reconstruction and novel view rendering. Regarding the hyperparameters used in our joint training of neural surface reconstruction and pose refinement, we have adopted settings consistent with those detailed in NeuS [1]. Specifically, we randomly sample 512 rays from a single image in each iteration, with 128 points sampled per ray.

A.4. Details for baseline methods

BARF [5] and L2G [30] are designed for LLFF and NeRF-Synthetic datasets. We find that naively running them with the original setting in DTU and MobileBrick scenes leads to divergence. Therefore, we tuned hyperparameters for them. Specifically, for BARF, we reduce the pose learning rate from the original 1e-3 to 1e-4. For L2G, we multiply the output of their local warp network by a small factor, *i.e.*, $\alpha = 0.01$. As SPARF [6] was tested in DTU scenes, we do not need to tune parameters. However, as it runs on sparse views (e.g., 3 or 6 views) as default, they build all-to-all correspondences. When running SPARF in dense views (up to 120 images), we build correspondences between neighbouring co-visible views (up to 45 degrees in the relative pose) considering the memory limit.

A.5. Details for experimental results

Quantitative ablation study results on DTU. Tab. 6 shows the quantitative results for different settings. Firstly, the comparison spanning from L1 to L4 serves to validate the effectiveness of the proposed components, which encompass critical elements like the PoRF (Pose Refinement) module and the epipolar geometry loss. This analysis highlights the significance of these components in enhancing the overall performance of the method. Secondly, the comparison encompassing L4 to L6 is intended to showcase the versatility and adaptability of the proposed method across different correspondence sources and scene representations. This demonstration underlines the method's capacity to deliver robust results regardless of variations in input data, thereby emphasising its universal applicability.

Qualitative reconstruction comparison. We include additional qualitative reconstruction results for both the DTU dataset in Fig. 5 and the MobileBrick dataset in Fig. 6. These visualisations serve to illustrate key points. It is evident that the reconstruction quality using COLMAP [4] pose information is marred by noise, making it challenging to enhance even when employing BARF [5]. SPARF [6] demonstrates improvements in visual quality, but it lags behind our method in terms of the precision and fidelity of the reconstructed object surfaces. These supplementary qualitative results underscore the superior performance and capabilities of our method in comparison to alternative approaches, particularly when it comes to achieving accurate and detailed object surface reconstructions.

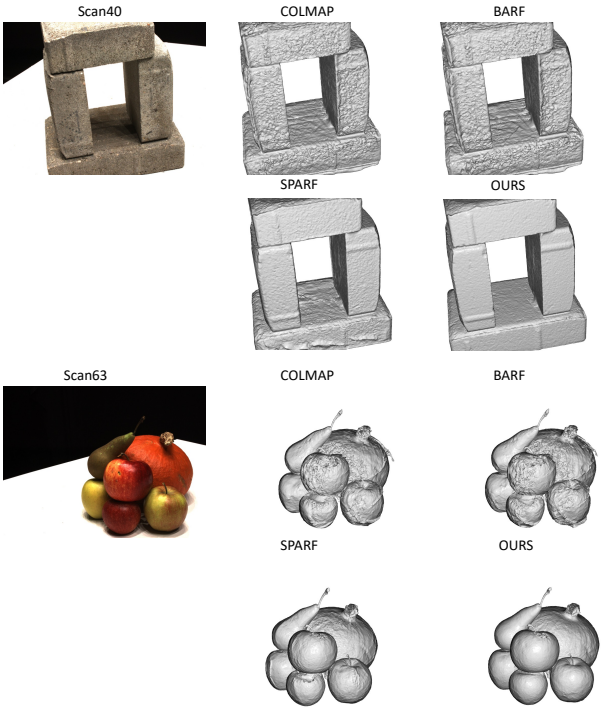


Figure 5. Qualitative reconstruction results on the DTU dataset. All meshes were generated by using Voxurf [2]. All refinement method takes the COLMAP [4] pose as the initial pose.

B. Additional experimental results

B.1. Novel view synthesis

We have also provided additional novel view synthesis results on both the DTU dataset (Table 8) and the MobileBrick dataset (Table 7). In these evaluations, we utilise Voxurf [2] for image rendering. It's essential to note that Voxurf was not originally designed for image rendering purposes, and it may face challenges, particularly when dealing with images that contain distant backgrounds, as observed in the MobileBrick dataset. These additional results offer a comprehensive view of our method's performance across different datasets, emphasising its strengths and limitations in novel view synthesis tasks. The qualitative comparison results are illustrated in Fig. 10 and Fig. 11, respectively.

Table 6. Quantitative ablation study results on DTU. The results are averaged over 15 test scenes. Baseline (B) denotes naive joint optimisation of pose parameters and NSR.

Methods	COLMAP	1. Baseline (B)	2. B+PoRF	3. B+L_{EG}	4. B+PoRF+L_{EG} (Ours)	5. Ours-LoFTR	6. Ours-NeRF
Rot. (Deg.)	0.65	0.38	0.19	0.20	0.14	**0.13**	0.14
Tran. (mm)	1.03	2.67	1.06	1.37	1.02	1.03	**1.00**

Table 7. Quantitative novel view synthesis results on the MobileBrick dataset. The Voxurf [2] is used for image rendering. The images in this dataset encompass 360-degree distant backgrounds, and it's worth noting that Voxurf was not specifically designed to render images and to handle challenges in such scenarios.

Poses	GT	ARKit	BARF [5]	SPARF [6]	OURS
PSNR ↑	23.65	17.94	**19.45**	18.87	18.97
SSIM ↑	0.719	0.561	**0.634**	0.607	0.619
LPIPS ↓	0.447	0.541	**0.501**	0.506	**0.501**

Table 8. Quantitative novel view synthesis results on the DTU dataset. The Voxurf [2] is used for reconstruction and image rendering.

Poses	GT	COLMAP	BARF [5]	SPARF [6]	OURS
PSNR ↑	31.84	27.63	27.50	29.59	**31.67**
SSIM ↑	0.931	0.861	0.856	0.892	**0.929**
LPIPS ↓	0.142	0.208	0.211	0.167	**0.143**

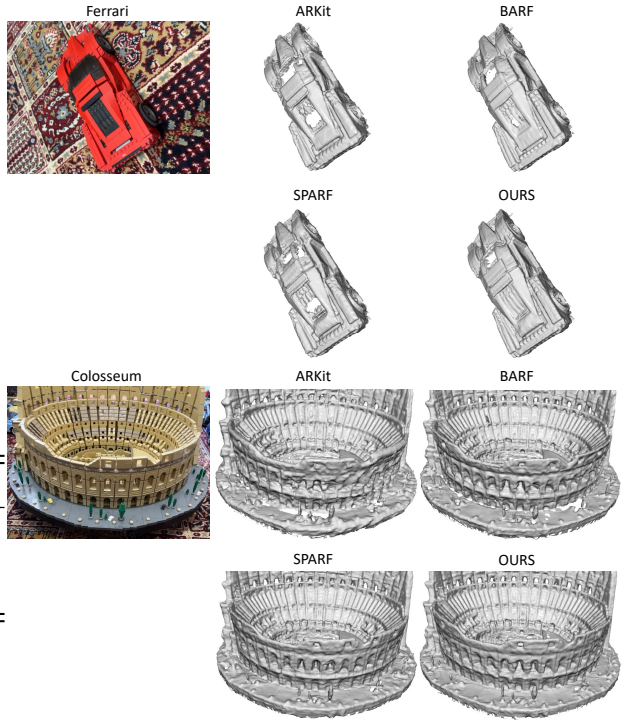


Figure 6. Qualitative reconstruction results on the MobileBrick dataset. All meshes were generated by using Voxurf [2]. All refinement method takes the ARKit pose as the initial pose.

B.2. Additional ablation studies

Ablation over the inputs of PoRF. In our PoRF module, the inputs comprise both the frame index and the initial camera pose. In this experiment, we conducted an ablation study to dissect the impact of each component on the results. Figure 7 presents the outcomes of this study, demonstrating that the removal of the initial camera pose results in a significant drop in performance, while the removal of the frame index causes a minor decrease. It's crucial to note that although the frame index appears to have a relatively low impact on accuracy, it remains an essential input. This is because the frame index is unique to each frame and serves a critical role in distinguishing frames that are closely situated in space.

B.3. Additional analysis

Impact of the pose noise on reconstruction. To understand how pose noises affect reconstruction accuracy, we make an empirical analysis by introducing different levels of pose noise on DTU [7]. The results are shown in Fig. 9, where we use Voxurf [2] to do reconstruction with different poses. To facilitate a comparison between rotation and translation, we use the COLMAP pose error as a reference, *i.e.*, using its pose error as a unit. It reveals that reconstruction accuracy is highly sensitive to rotation errors, whereas small perturbations in translation have minimal impact on reconstruction accuracy. Therefore, in the optimisation process, the reconstruction loss cannot provide strong supervision for optimising translation. This explains why our method shows significant improvement in rotation but minor improvement in translation.

Robustness to the initial pose noise. To assess the robustness of our method, we introduce varying degrees of noise to the ground truth (GT) pose during initialisation and subsequently apply our pose refinement process. The outcomes of this analysis are illustrated in Fig. 8, displaying quantitative results for the DTU dataset [7]. In line with the preceding analysis, we utilise the COLMAP pose error as a reference point for comparison. The findings underscore

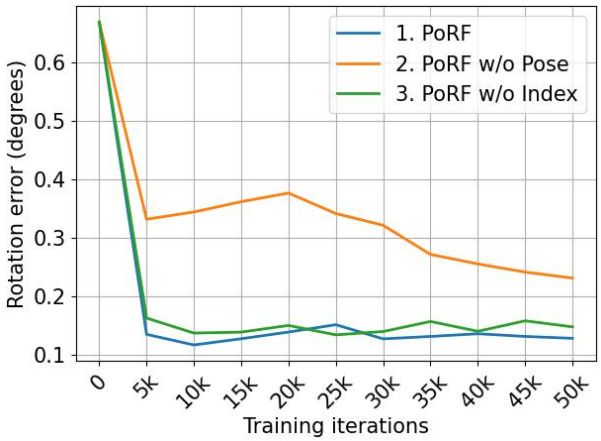


Figure 7. Ablation study results of the PoRF inputs on *Scan37*.

that our method consistently reduces pose error across different levels of pose noise, revealing its capacity for robust performance. Additionally, the method exhibits greater resilience to translation errors in comparison to rotation errors.

C. Limitations

First, our method builds upon the foundation of an existing technique, NeuS [1], for the joint optimisation of neural surface reconstruction and camera poses. Consequently, the training process is relatively slow, taking approximately 2.5 hours for 50,000 iterations. In future work, we plan to explore techniques like instant NGP [40] methods to accelerate this training phase further.

Second, it's essential to note that our method, unlike SPARF [6], is primarily designed for high-accuracy pose refinement. As a result, it requires nearly dense views and a robust camera pose initialisation for optimal performance. In practical real-world applications, achieving such dense views and quality camera pose initialisation can often be accomplished using existing methods like ARKit and COLMAP.

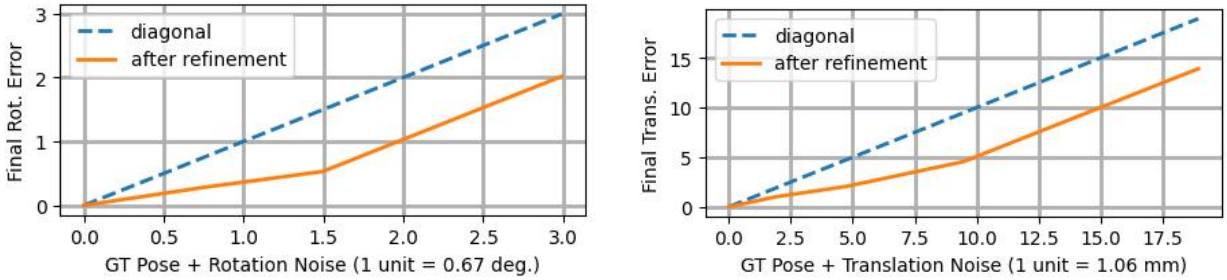


Figure 8. Robustness of our method to the initial pose noise on DTU (*Scan37*). The dashed lines and solid denote the pose error before and after refinement, respectively.

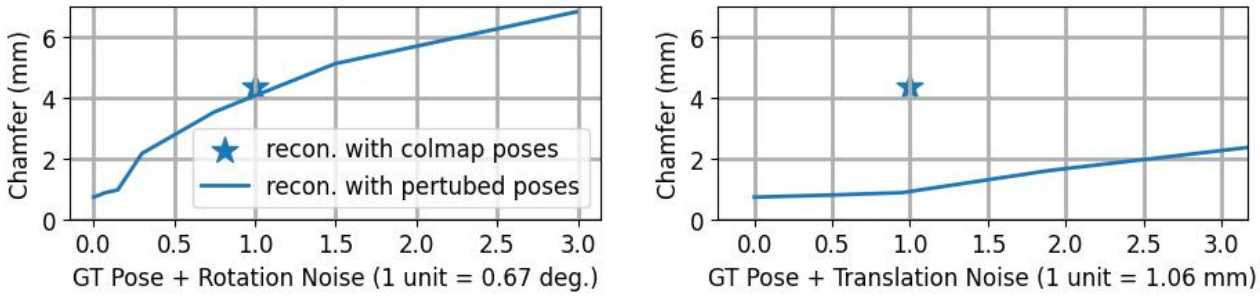


Figure 9. Reconstruction errors with pose errors on DTU (*Scan37*). We add noise to the GT pose and adopt the COLMAP pose error as one unit for comparison.



Figure 10. Qualitative novel view synthesis results on the DTU dataset. Images were rendered by using Voxurf [2].



Figure 11. Qualitative novel view synthesis results on the MobileBrick dataset. Images were rendered by using Voxurf [2].

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	PoRF: Pose Residual Field for Accurate Neural Surface Reconstruction
Publication Status	☑ Published □ Accepted for Publication □ Submitted for Publication □ Unpublished and unsubmitted work written in a manuscript style
Publication Details	Bian, J. W., **Bian, W.**, Prisacariu, V. A., & Torr, P. (2024). Porf: Pose residual field for accurate neural surface reconstruction. In *International Conference on Learning Representations (ICLR).*

Student Confirmation

Student Name:	Wenjing Bian		
Contribution to the Paper	• Joint conception of the idea • Running of baseline experiments • Research of prior work • Writing of the paper		
Signature	Wenjing Bian	Date	April 22, 2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Victor Adrian Prisacariu		
Supervisor comments The description is accurate		
Signature VPA	Date	April 22, 2025

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 7

Scene Coordinate Reconstruction Priors

Scene coordinate regression (SCR) models have show promising results in 3D vision tasks, including visual relocalisation and structure-from-motion (SfM). However, since SCR models typically require individual training for each scene, their performance can degrade when the input images provide insufficient multi-view constraints, leading to distortions and outliers in the reconstructed geometry, and ultimately affecting downstream tasks.

In this chapter, we propose two high-level reconstruction priors to mitigate such ambiguities by regularising the geometry of the predicted scene coordinates. The first is a hand-crafted prior that models the plausible depth distribution of the reconstructed scene coordinates or uses measured depth values when available. The second is a learned prior derived from a point cloud diffusion model trained to represent plausible indoor environments. We reformulate the SCR training process as a maximum likelihood estimation problem, which allows these priors to be seamlessly incorporated into existing SCR frameworks. These priors can be applied to state-of-the-art models such as ACE [44], GLACE [45], and ACE0 [41], resulting

in consistently improved 3D reconstruction quality. As a result, relocalisation accuracy and SfM registration rates are enhanced, without significantly increasing training time or affecting inference efficiency.

To summarise, our main contributions are:

- We reformulate SCR training as maximum likelihood learning, enabling the integration of reconstruction priors.
- We introduce two high-level geometric priors: one based on depth distributions and another learned from a diffusion model encoding plausible indoor geometry.
- We demonstrate that these priors improve scene representation quality, leading to better performance in downstream tasks including visual relocalisation and SfM.

In the following, we present our paper *Scene Coordinate Reconstruction Priors*, which is accepted by *International Conference on Computer Vision ICCV 2025* for conference publication.

Scene Coordinate Reconstruction Priors

Wenjing Bian[1,2] Axel Barroso-Laguna[1] Tommaso Cavallari[1]
Victor Adrian Prisacariu[1,2] Eric Brachmann[1]
[1]Niantic [2] University of Oxford

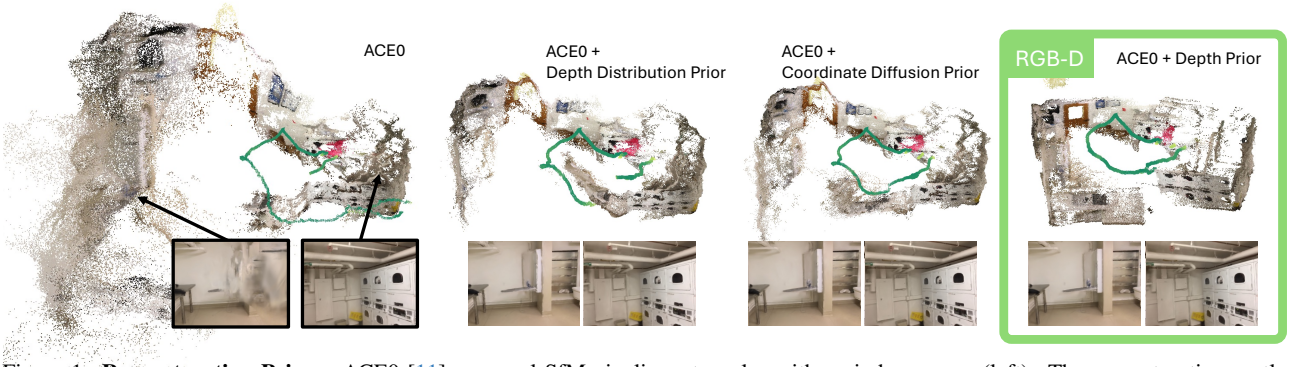


Figure 1. **Reconstruction Priors.** ACE0 [11], a neural SfM pipeline, struggles with an indoor scene (left). The reconstruction partly degenerates on the left side of the room, as is evident by the dispersed point cloud, camera poses (green) penetrating a wall, and artifacts in one of the views synthesized with Splatfacto [90] (bottom). We incorporate various priors into ACE0 which lead to a more consistent scene layout, better pose estimates, and fewer artifacts in synthesized views (center). For reference, we show a version of ACE0 taking RGB-D rather than RGB images as input (right).

Abstract

Scene coordinate regression (SCR) models have proven to be powerful implicit scene representations for 3D vision, enabling visual relocalization and structure-from-motion. SCR models are trained specifically for one scene. If training images imply insufficient multi-view constraints to recover the scene geometry, SCR models degenerate. We present a probabilistic reinterpretation of training SCR models, which allows us to infuse high-level reconstruction priors. We investigate multiple such priors, ranging from simple priors over the distribution of reconstructed depth values to learned priors over plausible scene coordinate configurations. For the latter, we train a 3D point cloud diffusion model on a large corpus of indoor scans. Our priors push predicted 3D scene points towards a more plausible geometry at each training step to increase their likelihood. On three indoor datasets our priors help learning better scene representations, resulting in more coherent scene point clouds, higher registration rates and better camera poses, with a positive effect on down-stream tasks such as novel view synthesis and camera relocalization.

1. Introduction

With the recent advent of learning-based structure-from-motion (SfM) pipelines, there are three avenues to further push their capabilities: 1) priors, 2) priors, and 3) priors.

In this work, we consider a particular class of neural SfM models that hitherto only incorporated very weak priors: Scene Coordinate Regression (SCR) [82]. SCR models are implicit scene representations. They are trained on images of a particular scene with known ground truth camera poses [3, 6, 7, 10, 82]. Once trained, SCR models allow to estimate the camera poses of unseen query images relative to the training scene, *i.e.* they enable visual relocalization. Recently, ACE0 [11] showed self-supervised training of SCR, turning it into a fully differentiable, neural SfM approach.

SCR reinterprets classical 3D reconstruction using a machine learning methodology. However, they still follow classical 3D vision principles when learning the 3D geometry of a scene: where feature-based methods triangulate sparse key points explicitly, SCR methods rely on dense, implicit triangulation of image patches [3]. Whether one triangulates implicitly or explicitly, without sufficient multi-view constrains triangulation fails [34]. This happens *e.g.*

for texture-poor areas, repetitive structures, reflections, *etc*. where multi-view observations cannot be resolved to the same 3D scene point. In Fig. 1, featureless walls and floor cause ACE0 to degenerate with an adverse effect on novel view synthesis on top of the ACE0 reconstruction.

Our aim is to overcome the ambiguities and potential incompleteness of scene-specific training data, by leveraging high-level reconstruction priors. Indeed, it is easy to see that the degenerate scene representation of Fig. 1 is unlikely to correspond to real scene geometry as the left half of the room is dispersed into space, and the estimated camera trajectory penetrates a wall. Following this intuition, we reformulate the training objective of SCR to easily incorporate reconstruction priors in a maximum likelihood framework. These priors can be relatively simple and hand-crafted or more involved and learned from data.

For hand-crafted priors, we model plausible distributions of depth values when reconstructing scene coordinates from RGB images, or condition the reconstruction on measured depth values if available. For a learned prior, we train a 3D diffusion model on point clouds of indoor scenes, to encode plausible scene geometries. Training generative models, like diffusion models, on 3D representations is a difficult endeavor. 3D datasets are orders of magnitude smaller than their 2D counterparts. The largest 3D datasets are comprised of simplistic 3D assets [18, 24, 25] wheres datasets with entire 3D scenes are smaller still, containing a few hundred scenes at most [1, 22, 48, 103]. We also still lack efficient but expressive architectures to generate entire 3D scenes with high fidelity. Both data and architecture limitations contribute to the fact that, so far, 3D point cloud diffusion was only shown for individual, isolated objects [56–58, 105]. We show that a lean 3D point cloud model, trained on relatively little data (around 700 indoor scenes) learns a coarse but useful prior over indoor scene layouts.

We summarize our **contributions**:

- We reformulate SCR training as maximum likelihood learning to enable incorporation of reconstruction priors.
- We propose various priors, including hand-crafted priors over the distribution of depth values of reconstructed scene coordinates, and a learned prior, which encodes plausible indoor scene layouts in a lean and efficient 3D point cloud diffusion model.
- We plug our priors into state-of-the-art SCR frameworks ACE [10], GLACE [93] and ACE0 [11]. They lead to learning better scene representations signified by more coherent and accurate point clouds, higher registration rates in SfM, and better pose estimates. Our approach does not significantly increase the training time of SCR models and it does not affect efficiency at query time.
- As a byproduct, we show how measured depth maps can be incorporated as a prior, leading to an effective RGB-D versions of ACE and ACE0.

2. Related Work

Visual Reconstruction and Relocalization Structure-from-motion refers to the problem of estimating the scene geometry and camera poses from a set of images [13, 64, 80, 83, 97]. Visual relocalization is a related task where the camera pose of a query image should be estimated relative to a set of mapping images with known camera poses [39, 73, 76, 78]. Both tasks have successfully been addressed with classical feature-matching where scenes are represented as sparse 3D point clouds. The current generation of learning-based feature matchers [29, 49, 74, 89] is very reliable [40], with latest methods like MicKey [2] and MASt3R [46, 96] demonstrating successful feature matching even for opposing shots. However, despite the robustness of feature matchers, classical triangulation-based methods rely on the availability of sufficient multi-view constraints to reconstruct a scene [34]. In texture-poor areas or with small camera baselines, point triangulation might still degenerate. Since feature-based pipelines consist of various processing stages with a complex control flow [35, 80, 83, 94, 97], it is difficult to infuse high-level priors for regularization.

Feed-forward reconstruction methods, like DUSt3R [96] and MASt3R [46], do suffer less from potential degeneracies of point triangulation. Trained on large datasets, these methods incorporate strong reconstruction priors to enable reconstruction in sparse-view scenarios with little to no visual overlap. However, these methods are inherently binocular, and are difficult to scale to large image collections. In a spirit similar to DUSt3R/MASt3R, our work aims at enabling strong priors for SCR methods.

As another alternative to feature matching, pose regression methods aim to predict the camera poses using a feed-forward network [12, 19, 41, 42, 81]. These networks represent the scene implicitly, and do not offer introspection on whether their scene representations are more or less prone to degeneracies. In any case, pose regression methods either lack accuracy [79], or depend on massive amounts of scene-specific synthesized training data [20, 61].

Scene Coordinate Regression (SCR) SCR models fuse the key point extraction and matching stages of classical feature-based methods into a single regression step, performed by a scene-specific machine learning model. Originally proposed by Shotton *et al*. [82] for relocalization in small-scale indoor scenes with RGB-D images and random forests, the approach was later extended to on-the-fly adaptation [15–17], RGB inputs [3, 7], using neural networks [5, 6, 8] and to larger scenes [4, 47, 93, 95]. Recently, the ACE framework [10] has demonstrated training of extremely memory-efficient SCR models in a few minutes per scene. Subsequently, ACE0 [11] extended the applicability

of ACE models from visual relocalization to structure-from-motion by demonstrating self-supervised training.

Most previous SCR methods rely entirely on scene-specific training, and make very little use of scene-agnostic data to distill reconstruction priors. Conversely, ACE and ACE0 use a pre-trained scene-agnostic feature encoder. However, as a low-level component, the feature encoder can do little to ensure coherence of the final scene representation. Marepo [21] pre-trains a scene-agnostic transformer to replace RANSAC [30] and PnP [33] in the ACE pipeline. In a similar spirit, SACReg [72] and SANet [101] pre-train scene-agnostic SCR predictors that learn to interpolate scene coordinate annotations of mapping images. Marepo, SACReg and SANet all concern test-time components of SCR pipelines, and assume that a valid scene representation has already been built. To the best of our knowledge, we are the first to regularizes SCR training by high-level priors, derived from scene-agnostic data.

Denoising Diffusion Models (DDMs) One of the priors we propose is a DDM on 3D point cloud to assess the likelihood of scene coordinate configurations. DDMs are generative models that incrementally transform random noise to a target distribution, *e.g*. transforming high-dimensional Gaussian noise to the distribution of natural images [37, 84, 85]. The transformation is implemented using a model that is trained to remove a small amount of noise at each timestep. Generation can be unconditional, or conditional, *e.g*. generating images of a pre-defined class [26, 36]. Due to the limited size and diversity of 3D datasets, diffusion for generation in 3D is difficult. 2D diffusion models have been utilized to regularize or guide 3D data generation [32, 66, 98]. Related to autonomous driving, large-scale scene generation using diffusion has been demonstrated for 2.5D LiDAR range images [63, 69, 106], and on coarse voxel grids [45, 52].

Diffusion models are related to score matching methods [86, 87] where models estimate the gradient of the log-likelihood of the target distribution [14, 37, 55]. This observation was used by DiffusioNeRF [99] to utilize a diffusion model to represent the prior probability over RGB-D image patches when training a neural radiance field [59]. We take inspiration from DiffusioNeRF's formulation but avoid the significant slow-down of rendering 2.5D image patches throughout training by regularizing directly in 3D.

DDMs for Point Clouds Luo and Hu [56] first demonstrated diffusion on 3D point clouds using a PointNet [67] architecture. Subsequent works [57, 60, 104, 105] proposed various pipeline improvements, including better backbone architectures like PointNet++ [68] and point-voxel CNN [54]. All these works are limited to modeling simplistic, isolated objects like *e.g*. those of ShapeNet [18]. Melas-Kyriazi *et al*. [58] show image-conditional diffusion of realistic, but still isolated, objects on Co3D [71]. To the best of our knowledge, generation of scene-level 3D point clouds using diffusion has not been demonstrated thus far. We use a point-voxel CNN [54] to model the distribution of scenes in ScanNet [22]. While our generative model cannot generate scenes with high fidelity, we show that it serves as an efficient prior for learning scene coordinate regression.

3. Method

SCR models [82] encode a scene into a scene-specific neural network f. The network f maps an image patch $\mathbf{p}_i$ centered around pixel i of image $\mathcal{I}$ to a 3D scene point $\mathbf{y}_i$,

$$\mathbf{y}_i = f(\mathbf{p}_i; \mathbf{w}), \tag{1}$$

where $\mathbf{w}$ denotes the parameters that encode the scene.

An explicit 3D point cloud of the scene can be extracted from f by running it over scene images and collecting the 3D points $\mathbf{y}_i$ [6]. However, the main application of SCR models is camera pose estimation, where f is applied to query images with unknown poses. Since the prediction $\mathbf{y}_i$ represents a 2D-3D correspondence from pixel i to scene space, the outputs of f can be used for camera pose estimation by feeding them into RANSAC [30] and PnP [33].

An SCR model f can be trained using RGB-D or RGB images [6]. Although we will present an RGB-D version of our approach, we are mainly interested in the general case and will assume that only RGB images are available if not stated otherwise. Training also requires known camera poses for the training images. Recently, ACE0 [11] has demonstrated that SCR models can be trained self-supervised where training camera poses are estimated by an SCR model of a previous reconstruction iteration.

With multiple images $\mathcal{I}_\text{M}$ and their known camera poses for a specific scene, f can be optimized using a reprojection loss L_reproj. However, when multi-view constraints from the images are insufficient or ambiguous, f may (partly) degenerate, and estimate scene points $\mathbf{y}_i$ that are noisy, distorted or plain outliers. Such artifacts in the scene coordinate prediction may lead to degraded performance in downstream tasks, such as SfM or relocalization.

To mitigate this, we reformulate the SCR training objective and interpret it as maximum likelihood learning. We incorporate general prior geometric constraints that can be applied during the training stage, and thus do not affect test time efficiency. We base our work on ACE [10], due to its good accuracy and attractive training time. We note that our priors can be readily applied to any derivative of ACE as long as it keeps the general training framework. We show results using ACE0 [11] for SfM, and GLACE [93] for relocalization. An overview of our system can be found in Fig. 2.

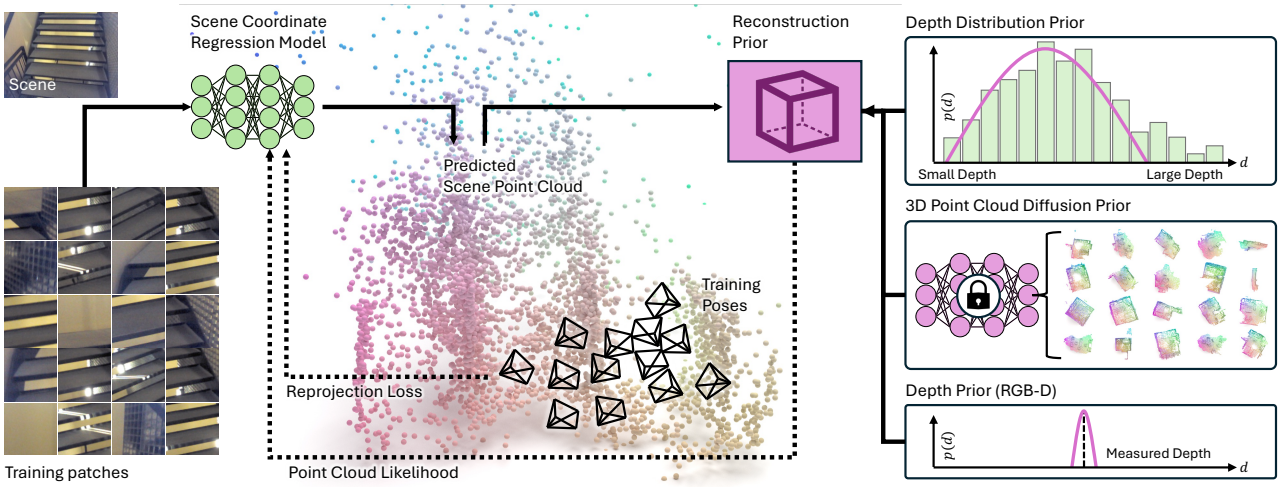


Figure 2. **System.** We train a scene coordinate regression (SCR) model to represent a scene. Following ACE [10] and ACE0 [11], the SCR network predicts scene coordinates for a batch of random image patches in each training iteration. We supervise the SCR network with the reprojection error of scene coordinates w.r.t. ground truth camera poses. Extending [10, 11], we introduce a reconstruction prior that outputs the gradient of the log-likelihood of the predicted point cloud. We investigate various priors: A *depth distribution prior* encourages depth values of reconstructed scene coordinates to follow a plausible distribution. A *3D point cloud diffusion prior* is a network that was trained offline on a large corpus of scenes and encodes plausible scene layouts. We can also infuse a *depth prior* if inputs are RGB-D.

3.1. ACE Framework

ACE decomposes a SCR model f into a scene-agnostic feature extractor f_{B} and a scene-specific regression head f_{H},

$$f(\mathbf{p}_i; \mathbf{w}) = f_{\text{H}}(\mathbf{f}_i; \mathbf{w}_{\text{H}}), \text{ with } \mathbf{f}_i = f_{\text{B}}(\mathbf{p}_i; \mathbf{w}_B). \quad (2)$$

The feature extractor f_{B} maps an image patch to a high-dimensional feature vector $\mathbf{f}_i$, and the regression head f_{H} maps the feature to a scene coordinate $\mathbf{y}_i$. Following ACE, we assume f_{B} was pre-trained and remains fixed. We focus on the mapping stage which optimizes f_{H}.

ACE first passes all mapping images $\mathcal{I}_{\text{M}}$ to the feature extractor f_{B} to obtain a training buffer with features for a large number of patches randomly sampled from mapping images. When training f_{H}, to achieve faster and more stable convergence, the buffered features are shuffled at each epoch. In each training iteration, a batch of N features $\mathbf{f} = \{\mathbf{f}_i \mid i = 1, ..., N\}$ is sampled from the feature buffer to estimate the 3D scene points $\mathbf{y} = \{\mathbf{y}_i \mid i = 1, ..., N\}$ for the corresponding 2D pixels from $\mathcal{I}_{\text{M}}$.

The ACE training loss $L_{\text{reproj}}(\mathcal{I}_{\text{M}}, \mathbf{y}, \mathbf{h}^*)$ is a pixel-wise projection loss applied to the 3D scene predictions $\mathbf{y}$ in each training iteration. Here, $\mathbf{h}^* = \{\mathbf{h}_i^* \mid i = 1, ..., N\}$ denotes the ground truth poses for each image from which the pixels are sampled. To regularize training in the first few iterations, ACE applies an additional loss, $L_{\text{init}}(\mathcal{I}_{\text{M}}, \mathbf{y}, \mathbf{h}^*)$, that mainly ensures that predicted coordinates are in front of their associated camera. For further details, please refer to ACE [10].

3.2. Probabilistic Training Objective

We recast the usual loss minimization objective of SCR as the maximization of a scene coordinate's probability. The posterior probability for the scene points $\mathbf{y}$, given the mapping images $\mathcal{I}_{\text{M}}$ and poses $\mathbf{h}^*$, is proportional to the product of the likelihood $p(\mathbf{h}^*, \mathcal{I}_{\text{M}} \mid \mathbf{y})$ and the prior $p(\mathbf{y})$:

$$p(\mathbf{y} \mid \mathbf{h}^*, \mathcal{I}_{\text{M}}) = \frac{p(\mathbf{h}^*, \mathcal{I}_{\text{M}} \mid \mathbf{y}) p(\mathbf{y})}{p(\mathbf{h}^*, \mathcal{I}_{\text{M}})}. \quad (3)$$

Taking the negative logarithm of the posterior and omitting the constant $p(\mathbf{h}^*, \mathcal{I}_{\text{M}})$ yields:

$$-\log p(\mathbf{y} \mid \mathbf{h}^*, \mathcal{I}_{\text{M}}) \propto -\log p(\mathbf{h}^*, \mathcal{I}_{\text{M}} \mid \mathbf{y}) - \log p(\mathbf{y}). \quad (4)$$

In ACE mapping, minimizing the reprojection error can be interpreted as maximizing the log-likelihood of the mapping views and poses given the predicted scene points, *i.e.*

$$-\log p(\mathbf{h}^*, \mathcal{I}_{\text{M}} \mid \mathbf{y}) := L_{\text{reproj}}. \quad (5)$$

Similarly, ACE's regularization term can be interpreted as a form of prior on scene coordinates, *i.e.* $-\log p(\mathbf{y}) := L_{\text{init}}$. However, ACE only optimizes either L_{reproj} or L_{init} per scene coordinate where the latter serves as an initialization target for the former. In contrast, our perspective suggests that reprojection error and prior should be optimized jointly to maximize the likelihood of a scene coordinate. In the following, we propose a set of alternative priors as stronger substitutes of L_{init}, and suitable to be jointly optimized with the reprojection error.

3.3. Depth Distribution Prior

Firstly, we present a prior that builds on, and extends, the intuition of ACE's original initialization loss. That loss ensures that the depth d_i of a predicted scene coordinate $\mathbf{y}_i$ is within sensible bounds, namely $0 < d_i < d_{\text{max}}$, where d_{max} is a user-defined upper bound. Between those bounds, the initialization loss provides no signal, but ACE switches to optimizing the reprojection error only.

Instead, we propose to model a continuous distribution of plausible depth values, and to encourage scene coordinate to adhere to that distribution. Real depth values are strictly positive, so a log-normal distribution would be a natural choice. However, we need our prior to be defined over the whole real line to be able to assess the likelihood of negative depth values which can occur during SCR training. We empirically found a Laplacian distribution, $\text{Lap}(d \mid \mu, b)$, to be a good choice where we fit mean μ and bandwidth b on a held out set of scenes with measured (*i.e.* ground truth) depth values.

We can directly use the negative log-likelihood of individual depth values as a prior per pixel i, *i.e.*

$$\log p(\mathbf{y_i}) \coloneqq \lambda_{\text{reg}} \ \log \text{Lap}(d_i \mid \mu, b), \tag{6}$$

where λ_{reg} is a hyper-parameter to balance the prior with the reprojection loss. This prior corresponds to an L_1 loss on the predicted scene coordinate's depth, pulling it to the empirical mean.

As a slightly more advanced prior, we could try to make sure that the set of all predicted scene coordinates have depth values that follow the distribution of depth values in a real scene. To this end, we utilize a distribution loss between the predicted depth values and our prior. We take advantage of the ACE framework which constructs mini-batches associated with random scene points throughout optimization. Therefore, we can assume that the distribution of predicted depth values of a mini-batch roughly follows the depth distribution of the entire scene. Consequently, an alternative prior is to use the Wasserstein distance between the set of predicted depth values of a mini-batch, $\{d_i\}$, and the target Laplace distribution:

$$\log p(\mathbf{y_i}) \coloneqq \lambda_{\text{reg}} \ W\left[\{d_i\}, \ \text{Lap}(\cdot \mid \mu, b)\right]. \tag{7}$$

3.4. Depth Prior (RGB-D)

The log-likelihood prior of Eq. 6 lends itself to an extension that ingests measured depth values d_i^* if available. In this case, we substitute the broad distribution over plausible depth values with a narrow distribution centered at the measured depth value for each pixel:

$$\log p(\mathbf{y_i}) \coloneqq \lambda_{\text{reg}} \ \log \text{Lap}(d_i \mid d_i^*, b'). \tag{8}$$

Here, the bandwidth b' is a user-defined value that controls the tolerance of the prior. We will use $b' = 10\text{cm}$.

3.5. Point Cloud Diffusion Prior

As a final prior, we propose adding a 3D diffusion prior to SCR training to regularize the geometry of the 3D scene points. The prior is trained on a held out set of scenes, and encodes knowledge about plausible scene point clouds.

In general forward diffusion, Gaussian noise ϵ is progressively added to an input signal $\mathbf{x}_0$ over T time steps until the signal becomes completely noisy. In contrast, SCR training can be regarded as incrementally recovering the scene point cloud from a random initialization. Therefore, SCR training aligns with a reversed diffusion process, and a 3D diffusion model can be used to guide SCR optimization.

Diffusion models [37, 85] learn to recover the original signal $\mathbf{x}_0$ by estimating the noise added to the noisy signal $\mathbf{x}_\tau$ at time step $\tau \in [0, T-1]$, through a neural network ϵ_θ. It has been shown in [37, 91, 99] that the noise estimation is proportional to the score function of the input signal, *i.e.*

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}) \coloneqq -\lambda_{\text{reg}} \epsilon_\theta(\mathbf{x}_\tau, \tau). \tag{9}$$

In our context, this allows us to approximate the prior over 3D coordinates $\mathbf{y}$ with a diffusion denoising model ϵ_θ. This prior is inspired by DiffusioNeRF [99] with the main difference that we regularize directly in 3D, avoiding a significant slowdown when regularizing RGB-D renderings as in [99].

Architecture Our noise estimator $\epsilon_\theta(\mathbf{x}_\tau, \tau)$ takes the noisy point cloud $\mathbf{x}_\tau \in \mathbb{R}^{N \times 3}$ and the diffusion timestep τ as inputs. We adopt PVCNN [53] with certain modifications made by [58] for diffusion timestep embedding.

Training We follow the training protocol in DDPM [37]. During each forward iteration, we first normalize the input point cloud $\mathbf{x}_0$ with a predefined scaling factor to ensure the points are within the range $[-1, 1]$. We then transform $\mathbf{x}_0$ to a noisy version $\mathbf{x}_\tau$ by adding noise to $\mathbf{x}_0$ according to the noise schedule at a randomly sampled time step τ. The training objective of the diffusion model is expressed as:

$$L_{\text{training}} = \mathbb{E}_{\tau, \epsilon \sim \mathcal{N}(0,1), \mathbf{x}_0} \left[\left\| (\epsilon - \epsilon_\theta(\mathbf{x}_\tau, \tau)) \right\|_2^2 \right]. \tag{10}$$

Inference Once the diffusion model is trained, we integrate it into SCR mapping. We take the estimated scene points $\mathbf{y}$ as input for the diffusion model and estimate the noise. We use the estimate as regularization as specified by Eq. (9). The distributions of the Gaussian noise and the noise encountered during ACE mapping differ, particularly during the first mapping iterations. Therefore, we apply the diffusion prior only after iteration 5k of ACE mapping. We align the diffusion time $\frac{T}{20}$ with ACE iteration 5k, and linearly interpolate timestep τ down to 0 as training progresses. We do not apply the diffusion prior to points with a reprojection error smaller than 30 pixels, as we assume that sufficient multi-view constraints exist for those points.

Table 1. **Reconstruction of ScanNet [22] and Indoor6 [27].** We compare results of ACE0 without and with our priors added. We report the percentage of images registered to the reconstruction (*Reg. Rate*), absolute trajectory and relative pose errors (*ATE*/*RPE*) as well as median pose errors compared to pseudo ground truth (pGT), and PSNR of novel view synthesis using two separate evaluation splits: the usual split with one test frame / seven training frames, and a harder split alternating between 60 test frames and 60 training frames.

		Comparison to Pose pGT (BundleFusion)				Splatfacto PSNR (dB)		Indoor6	
		Reg. Rate	ATE (cm)	RPE (cm)	Med. Err. (cm/°)	1/7	60/60	Reg. Rate	PSNR (dB)
RGB	ACE0	98.1%	26.6	4.0	19.7 / 9.0	30.2	22.3	57.1%	13.5
	ACE0 + Laplace NLL (Ours)	**98.9%**	25.4	**3.5**	17.5 / 8.8	30.2	22.2	58.0%	14.1
	ACE0 + Laplace WD (Ours)	98.7%	25.9	3.6	17.5 / 6.8	30.3	21.7	57.7%	14.1
	ACE0 + Diffusion (Ours)	98.6%	26.5	3.8	18.8 / 8.9	30.2	22.4	**61.8%**	**14.6**
RGB-D	ACE0 + DSAC* Loss	96.2%	29.2	6.0	20.9 / 5.9	30.0	21.9	N/A	N/A
	ACE0 + Laplace NLL (Ours)	**98.9%**	**18.3**	**3.5**	**12.8 / 4.4**	**30.6**	**22.9**	N/A	N/A

4. Experiments

Implementation Details We implement our approach in Pytorch [65], based on the official implementation of ACE0 [11]. Unless otherwise specified, we use the same parameter settings as ACE0 [11] for reconstruction experiments, and the same parameters as ACE [10] for relocalization experiments. We also integrate the diffusion prior into GLACE [93]. In this variant, we maintain the regularization configuration identical to its implementation in ACE, while keeping all other settings unchanged from the original GLACE configuration.

Training the Priors We fit our priors to the training set of ScanNetV2 [22] which consists of 706 scenes, densely reconstructed from RGB-D images. For the depth distribution prior (Sec. 3.3), we fit a Laplace distribution to randomly sampled depth values of the training images, yielding a mean of $\mu = 1.73$m and a bandwidth of $b = 60$cm. For the diffusion model (Sec. 3.5), we use the ground truth point cloud of each training scene as target. We train the diffusion model with a batch size of 16 for 100,000 iterations on a single V100 GPU. We use AdamW [44], with a learning rate that decays linearly from 0.0002 to 0 throughout the training process. In each forward iteration, we randomly sample 5120 points from a point cloud in the training dataset and apply augmentations including random rotation, translation and scaling within a certain range. The total number of diffusion time steps is set to 200.

4.1. Structure-from-Motion

We reconstruct a number of indoor scenes using ACE0 [11], with and without incorporating our priors.

Datasets We evaluate on the first 20 test scenes of ScanNetV2 [22], and on the mapping sequences of Indoor6 [27]. Where ScanNet consists of single room scans, Indoor6 features six larger indoor environments comprised of multiple rooms and is therefore much harder to reconstruct for ACE0. Indoor6 does not come with RGB-D images.

Metrics We report the percentage of images successfully registered to the reconstruction (*Reg. Rate*), *i.e.* images with a final inlier count above 1,000 [11]. We confirm the quality of estimated camera poses using novel view synthesis [11, 92]. That is, we estimate camera poses using all images. Then, we divide images into training and validation images for Splatfacto [43, 90, 102] to check whether we can re-render the scene based on the estimated camera poses. We report *PSNR* numbers on the usual training/validation split of Splatfacto, taking one validation image for every seven training images. For ScanNet we also report results for a harder split, alternating between 60 validation images and 60 training images. Finally, for ScanNet, we compare the estimated poses to the pseudo ground truth (pGT) that comes with the dataset, estimated by BundleFusion [23], a RGB-D SLAM system. While the accuracy limits of the pGT are unknown [9], we assume that it is sufficient to gauge the global consistency of our estimated poses. To this end, we report the absolute trajectory error (ATE) and relative poses errors (RPE) [88], as well as median rotation and translation errors. These metrics assume least-squares alignment of the estimated camera trajectory to the pGT camera trajectory.

Discussion We report results in Table 1. We couple our Laplace depth distribution prior (Sec. 3.3) with log-likelihood optimization (*Laplace NLL*) as well as the Wasserstein distance (*Laplace WD*). Both variations of the prior increase the registration rates and pose quality on ScanNet. The novel view synthesis quality is largely on par with ACE0, except for a noticeable drop in PSNR for the 60/60 evaluation split when using the Wasserstein distance. Effects are stronger on the Indoor6 dataset where ACE0 struggles due to the size of the environments. Both depth distribution priors lead to higher registration rates, and increase PSNR by 0.6 dB.

The diffusion prior yields slight but consistent improvements across all metrics on ScanNet. On Indoor6, the diffusion prior causes a significant increase in registered im-

Table 2. **Relocalization Results on 7Scenes [82].** We report the percentage of test images below a 5cm/5° pose error, mapping time and map size. Methods in "SCR w/ 3D" use depth or 3D point cloud supervision during mapping. Best results within the SCR groups are highlighted in **bold**. All methods use SfM mapping poses.

Type	Method	Mapping Time	Map Size	Chess	Fire	Heads	Office	Pumpkin	Redkitchen	Stairs	Avg
FM	AS (SIFT) [77]	∼1.5h	∼200MB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	**98.5%**
	D.VLAD+R2D2 [38]	∼1.5h	∼1GB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	95.7%
	hLoc (SP+SG) [73, 74]	∼1.5h	∼2GB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	95.7%
SCR w/ 3D	DSAC* [6]	15h	28MB	99.8%	98.9%	99.8%	98.5%	98.9%	97.8%	93.8%	**98.2%**
	SRC [28]	2min	40MB	89.1%	79.6%	97.6%	84.1%	65.7%	87.3%	64.7%	81.1%
	FocusTune [62]	6min	4MB	99.7%	99.0%	87.1%	99.9%	**99.9%**	**100%**	**99.5%**	97.9%
	ACE [10, 11] + DSAC* Loss [6]	5.5min	4MB	**100%**	**99.5%**	**100%**	98.8%	96.2%	98.2%	82.3%	96.4%
	ACE + Laplace NLL (Ours)	5.5min	4MB	**100%**	99.2%	**100%**	**99.8%**	97.2%	98.4%	87.3%	97.4%
SCR	DSAC* [6]	15h	28MB	**100%**	99.1%	98.8%	99.3%	99.4%	96.9%	78.6%	96.0%
	GLACE [93]	6min	9MB	**100%**	**99.9%**	99.9%	99.8%	**100%**	98.2%	71.2%	95.6%
	GLACE + Diffusion (Ours)	9min	9MB	**100%**	99.8%	**100%**	99.9%	99.5%	98.8%	73.6%	95.9%
	ACE [10]	5min	4MB	**100%**	99.5%	99.7%	**100%**	99.9%	98.6%	81.9%	97.1%
	ACE + Laplace NLL (Ours)	4.5min	4MB	**100%**	98.9%	99.9%	99.9%	**100%**	98.5%	84.2%	97.3%
	ACE + Laplace WD (Ours)	4.5min	4MB	**100%**	99.6%	**100%**	99.9%	99.8%	98.2%	83.2%	97.2%
	ACE + Diffusion (Ours)	8min	4MB	**100%**	99.5%	**100%**	**100%**	99.0%	**99.1%**	**86.2%**	**97.7%**

ages (+4.7%) leading to noticeable higher PSNR (+1.1 dB). The diffusion prior takes the global scene layout into account, providing a stronger regularization signal than our depth distribution priors.

On ScanNet, we also test a version of ACE0 with our RGB-D prior (Sec. 3.4). We compare to an ACE0 RGB-D baseline where we activate the ACE0 RGB-D loss, usually only applied when initializing the reconstruction, throughout the entire reconstruction. The ACE0 RGB-D loss is inspired by DSAC* [6] and optimizes the distance of predicted scene coordinates to ground truth derived from depth maps. The loss switches to the reprojection errors, when predicted scene coordinates are within 10cm of the ground truth. This RGB-D baseline of ACE0 (denoted *ACE0 + DSAC* Loss*) performs worse than the default ACE0 on ScanNet, presumably due to the hard switch between RGB and RGB-D terms reducing optimization stability. In contrast, ACE0 coupled with our depth prior, derived from our probabilistic interpretation of SCR optimization, yields significant improvements across all metrics.

4.2. Relocalization Results

We evaluate performance of our priors on the relocalization task using the 7Scenes and Indoor6 datasets, comparing it against other Scene Coordinate Regression (SCR) methods and feature-matching (FM) approaches. We report the percentage of images localized within a 5cm/5° threshold [82].

7Scenes Results in Tab. 2 demonstrate that ACE coupled our priors achieve higher relocalization accuracy than ACE alone. The diffusion prior has the strongest effect, improving results by 4.1% on the most difficult scene, *Stairs*. The supplement includes results using alternative ground truth poses [9] where the same trend holds. In Fig. 3 we show qualitatively how the diffusion prior leads to a more compact representation for this scene. To underline the versatility of our priors, we report results of GLACE [93], a recent ACE extension, coupling with our diffusion prior. Again, relocalisation results improve on average, particularly on the Stairs scene. In the supplement, we analyze the effect of our priors on the point clouds learned by ACE.

Table 3. **Relocalization Results on Indoor6 [27].** We report relocalization accuracy (5cm, 5°) and mapping time. Except for EGFS, we report all results as mean over 5 runs. Best results **bold** within each pair w/ and w/o our diffusion prior (denoted *-Diff*).

		Reloc. Acc.	Map. Time
N=5120	EGFS [51]	56.1%	21 min
	GLACE [93]	44.2% ± 1.8%	11 min
	GLACE-Diff (Ours)	**46.4%** ± 1.9%	15 min
	ACE [10]	36.2% ± 1.5%	5 min
	ACE-Diff (Ours)	**37.5%** ± 1.8%	8 min
N=51200	GLACE [93]	69.5% ± 1.4%	33 min
	GLACE-Diff (Ours)	**69.6%** ± 2.0%	40 min
	ACE [10]	57.2% ± 1.6%	10 min
	ACE-Diff (Ours)	**57.9%** ± 1.1%	13 min

In terms of mapping time, our depth distribution priors lead to simplified optimization objectives compared to ACE, leading to a slight reduction in mapping time. On the other hand, incorporating the diffusion prior incurs additional cost, as we need to execute a separate model many times throughout ACE mapping. However, the nominal increase in mapping time is still modest, with +3 minutes.

Indoor6 In Tab. 3, we report the average relocalization accuracy on Indoor6 for ACE [10]; GLACE [93] with and without our diffusion prior. Our experiments show that increasing the batch size significantly improves the local-

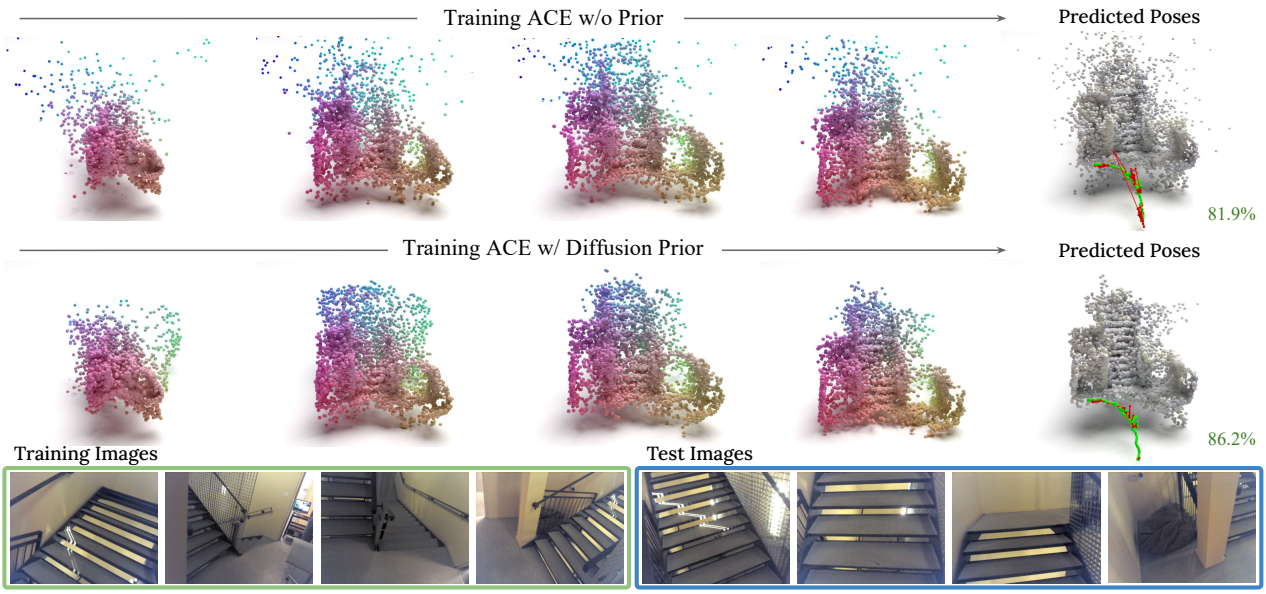


Figure 3. **Diffusion Prior on Stairs.** We guide the training of ACE [10] on the Stairs scene with our 3D diffusion model, leading to a more coherent scene geometry, and higher pose estimation accuracy on test images (5cm,5° threshold).



Figure 4. **Generated Point Clouds.** Point clouds generated by our diffusion model together with ScanNet point clouds used for training.

ization accuracy, presumably due to the larger size of the scenes. Therefore, we evaluated two versions of each method, using batch sizes of 5,120 and 51,200. We observed noticeable variance in relocalization accuracy on this dataset, and report mean results over 5 runs. Adding our prior helps on average, although ultimate conclusions are difficult due to the variance in this dataset. For per-scene results, please see the supplement. For EGFS [51], another ACE derivative, the code is not publicly available yet. We report results with batch size 5,120 from their paper. We note that EGFS could be coupled with our diffusion prior, as well.

4.3. Prior Introspection

Fig. 4 shows examples of point clouds generated by the diffusion model, compared with real ScanNet scenes, illustrating the geometric prior learned by the model. Although the model is unable to synthesize fine details, it represents sensible room layouts sufficient to regularize SCR training. We include further analysis in the supplement, regarding point cloud encoder architecture, efficiency, the prior's effect on scarce data and hyper-parameter choices.

5. Conclusion

We have presented a probabilistic reformulation of scene coordinate regression training that allows for the easy incorporation of reconstruction priors. We presented multiple potential priors: priors regularizing the distribution of reconstructed depth values, a prior leveraging measured depth, as well as a high-level learned prior in the form of a 3D point cloud diffusion model. We found that the regularization terms can be integrated into ACE-based frameworks, such as ACE, ACE0 or GLACE, yielding notable performance gains while not affecting the test-time latency.

Limitations Our experiments have focused on indoor scenes. Extending our approach to outdoor scenes would require more involved models of depth distribution, and more diverse data for diffusion training to learn a comprehensive prior over larger areas. A point cloud encoding network with better expressiveness could result in generations with higher fidelity, but it would need to remain efficient to be practical. An additional conditional signal for the diffusion model can also be a promising direction for future research, but would also require more training data to generalize.

References

[1] Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Áron Monszpart, Victor Adrian Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *ECCV*, 2022. 2

[2] Axel Barroso-Laguna, Sowmya Munukutla, Victor Prisacariu, and Eric Brachmann. Matching 2d images in 3d: Metric relative pose from metric correspondences. In *CVPR*, 2024. 2

[3] Eric Brachmann and Carsten Rother. Learning Less is More - 6D Camera Localization via 3D Surface Regression. In *CVPR*, 2018. 1, 2

[4] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *ICCV*, 2019. 2

[5] Eric Brachmann and Carsten Rother. Neural- Guided RANSAC: Learning where to sample model hypotheses. In *ICCV*, 2019. 2

[6] Eric Brachmann and Carsten Rother. Visual camera relocalization from RGB and RGB-D images using DSAC. *IEEE TPAMI*, 2021. 1, 2, 3, 7

[7] Eric Brachmann, Frank Michel, Alexander Krull, Michael Y. Yang, Stefan Gumhold, and Carsten Rother. Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image. In *CVPR*, 2016. 1, 2

[8] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC-differentiable ransac for camera localization. In *CVPR*, 2017. 2

[9] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In *ICCV*, 2021. 6, 7, 1

[10] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *CVPR*, 2023. 1, 2, 3, 4, 6, 7, 8

[11] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. In *ECCV*, 2024. 1, 2, 3, 4, 6, 7

[12] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-Aware Learning of Maps for Camera Localization. In *CVPR*, 2018. 2

[13] Matthew Brown and David G. Lowe. Unsupervised 3D object recognition and reconstruction in unordered datasets. In *3DIM*, 2005. 2

[14] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *ECCV*, 2020. 3

[15] Tommaso Cavallari, Stuart Golodetz, Nicholas A Lord, Julien Valentin, Luigi Di Stefano, and Philip HS Torr. On-the-fly adaptation of regression forests for online camera relocalisation. In *CVPR*, 2017. 2

[16] Tommaso Cavallari, Luca Bertinetto, Jishnu Mukhoti, Philip Torr, and Stuart Golodetz. Let's take this online: Adapting scene coordinate regression network predictions for online rgb-d camera relocalisation. In *3DV*, 2019.

[17] Tommaso Cavallari, Stuart Golodetz, Nicholas A. Lord, Julien Valentin, Victor A. Prisacariu, Luigi Di Stefano, and Philip H. S. Torr. Real-time rgb-d camera pose estimation in novel scenes using a relocalisation cascade. *IEEE TPAMI*, 2019. 2

[18] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 2, 3

[19] Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-PoseNet: Absolute pose regression with photometric consistency. In *3DV*, 2021. 2

[20] Shuai Chen, Xinghui Li, Zirui Wang, and Victor Prisacariu. Dfnet: Enhance absolute pose regression with direct feature matching. In *ECCV*, 2022. 2

[21] Shuai Chen, Tommaso Cavallari, Victor Adrian Prisacariu, and Eric Brachmann. Map-relative pose regression for visual re-localization. In *CVPR*, 2024. 3

[22] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 2, 3, 6, 1

[23] Angela Dai, Matthias Nießner, Michael Zollöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration. *ACM TOG*, 2017. 6

[24] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv*, 2023. 2

[25] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023. 2

[26] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 3

[27] Tien Do, Ondrej Miksik, Joseph DeGol, Hyun Soo Park, and Sudipta N Sinha. Learning to detect scene landmarks for camera localization. In *CVPR*, 2022. 6, 7

[28] Siyan Dong, Shuzhe Wang, Yixin Zhuang, Juho Kannala, Marc Pollefeys, and Baoquan Chen. Visual localization via few-shot scene region classification. In *3DV*, 2022. 7, 2

[29] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust Dense Feature Matching. In *CVPR*, 2024. 2

[30] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications

to image analysis and automated cartography. *Communications of the ACM*, 1981. 3

[31] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018. 1

[32] Ruiqi Gao*, Aleksander Holynski*, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul P. Srinivasan, Jonathan T. Barron, and Ben Poole*. Cat3d: Create anything in 3d with multi-view diffusion models. In *NeurIPS*, 2024. 3

[33] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE TPAMI*, 2003. 3

[34] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003. 1, 2

[35] Xingyi He, Jiaming Sun, Yifan Wang, Sida Peng, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Detector-free structure from motion. In *CVPR*, 2024. 2

[36] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshops*, 2021. 3

[37] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 3, 5, 1

[38] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Vincent Leroy, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, and Gabriela Csurka. Robust image retrieval-based visual localization using kapture. *arXiv*, 2020. 7, 2

[39] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, Vincent Leroy, and Gabriela Csurka. Robust image retrieval-based visual localization using Kapture, 2020. 2

[40] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image Matching across Wide Baselines: From Paper to Practice. *IJCV*, 2020. 2

[41] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017. 2

[42] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-DOF camera relocalization. In *CVPR*, 2015. 2

[43] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42(4), 2023. 6

[44] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv*, 2014. 6

[45] Jumin Lee, Woobin Im, Sebin Lee, and Sung-Eui Yoon. Diffusion probabilistic models for scene-scale 3d categorical data. *arXiv*, 2023. 3

[46] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r. In *ECCV*, 2024. 2

[47] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *CVPR*, 2020. 2

[48] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 2

[49] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023. 2

[50] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE TPAMI*, 2015. 1

[51] Ting-Ru Liu, Hsuan-Kung Yang, Jou-Min Liu, Chun-Wei Huang, Tsung-Chih Chiang, Quan Kong, Norimasa Kobori, and Chun-Yi Lee. Reprojection errors as prompts for efficient scene coordinate regression. In *ECCV*, 2025. 7, 8, 1

[52] Yuheng Liu, Xinke Li, Xueting Li, Lu Qi, Chongshou Li, and Ming-Hsuan Yang. Pyramid diffusion for fine 3d large scene generation. In *ECCV*, 2024. 3

[53] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. *NeurIPS*, 2019. 5, 2

[54] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *NeurIPS*, 2019. 3, 4

[55] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv*, 2022. 3

[56] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *CVPR*, 2021. 2, 3, 4

[57] Zhaoyang Lyu, Zhifeng Kong, Xudong Xu, Liang Pan, and Dahua Lin. A conditional point diffusion-refinement paradigm for 3d point cloud completion. In *ICLR*, 2022. 3

[58] Luke Melas-Kyriazi, Christian Rupprecht, and Andrea Vedaldi. Pc2: Projection-conditioned point cloud diffusion for single-image 3d reconstruction. In *CVPR*, 2023. 2, 3, 5

[59] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 3

[60] Shentong Mo, Enze Xie, Ruihang Chu, Lanqing HONG, Matthias Nießner, and Zhenguo Li. Dit-3d: Exploring plain diffusion transformers for 3d shape generation. In *NeurIPS*, 2023. 3

[61] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. LENS: Localization enhanced by nerf synthesis. In *CoRL*, 2021. 2

[62] Son Tung Nguyen, Alejandro Fontan, Michael Milford, and Tobias Fischer. Focustune: Tuning visual localization through focus-guided sampling. In *WACV*, 2024. 7

[63] Lucas Nunes, Rodrigo Marcuzzi, Benedikt Mersch, Jens Behley, and Cyrill Stachniss. Scaling Diffusion Models to Real-World 3D LiDAR Scene Completion. In *CVPR*, 2024. 3

[64] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes Lutz Schönberger. Global structure-from-motion revisited. In *ECCV*, 2024. 2

[65] 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. 6

[66] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2023. 3

[67] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 3

[68] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 3

[69] Haoxi Ran, Vitor Guizilini, and Yue Wang. Towards realistic scene generation with lidar diffusion models. In *CVPR*, 2024. 3

[70] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 1

[71] 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 *ICCV*, 2021. 3

[72] Jerome Revaud, Yohann Cabon, Romain Brégier, JongMin Lee, and Philippe Weinzaepfel. Sacreg: Scene-agnostic coordinate regression for visual localization. In *CVPRW*, 2024. 3

[73] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 2, 7

[74] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2, 7

[75] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. Back to the feature: Learning robust camera localization from pixels to pose. In *CVPR*, 2021. 2

[76] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *ICCV*, 2011. 2

[77] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *PAMI*, 2016. 7, 2

[78] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. *IEEE TPAMI*, 2017. 2

[79] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *CVPR*, 2019. 2

[80] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2

[81] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *ICCV*, 2021. 2

[82] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *CVPR*, 2013. 1, 2, 3, 7

[83] Noah Snavely, Steven Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM TOG*, 2006. 2

[84] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 3

[85] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 3, 5

[86] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019. 3

[87] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 3

[88] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IROS*, 2012. 6

[89] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *CVPR*, 2021. 2

[90] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *SIGGRAPH*, 2023. 1, 6

[91] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 2011. 5

[92] Michael Waechter, Mate Beljan, Simon Fuhrmann, Nils Moehrle, Johannes Kopf, and Michael Goesele. Virtual rephotography: Novel view prediction error for 3d reconstruction. *ACM TOG*, 2017. 6

[93] Fangjinhua Wang, Xudong Jiang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. Glace: Global local accelerated coordinate encoding. In *CVPR*, 2024. 2, 3, 6, 7, 1

[94] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *CVPR*, 2024. 2

[95] Shuzhe Wang, Zakaria Laskar, Iaroslav Melekhov, Xiaotian Li, Yi Zhao, Giorgos Tolias, and Juho Kannala. Hscnet++ : Hierarchical scene coordinate classification and regression for visual localization with transformer. *IJCV*, 2024. 2

[96] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 2

[97] Changchang Wu. Towards linear-time incremental structure from motion. In *3DV*, 2013. 2

[98] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P. Srinivasan, Dor Verbin, Jonathan T. Barron, Ben Poole, and Aleksander Holynski. ReconFusion: 3D Reconstruction with Diffusion Priors . In *CVPR*, 2024. 3

[99] Jamie Wynn and Daniyar Turmukhambetov. DiffusioNeRF: Regularizing Neural Radiance Fields with Denoising Diffusion Models. In *CVPR*, 2023. 3, 5, 1

[100] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *ICCV*, 2019. 1

[101] Luwei Yang, Ziqian Bai, Chengzhou Tang, Honghua Li, Yasutaka Furukawa, and Ping Tan. Sanet: Scene agnostic network for camera localization. In *ICCV*, 2019. 3, 2

[102] 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*, 2024. 6

[103] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023. 2

[104] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. In *NeurIPS*, 2022. 3

[105] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *ICCV*, 2021. 2, 3

[106] Vlas Zyrianov, Xiyue Zhu, and Shenlong Wang. Learning to generate realistic lidar point cloud. In *ECCV*, 2022. 3

Scene Coordinate Reconstruction Priors

Supplementary Material

6. Implementation Details

6.1. Diffusion Training

The point clouds in ScanNetV2 [22] are confined to the positive octant of the coordinate system, with the xy-plane aligned to the floor. To enhance the diversity of the training data, we first re-center the point clouds by shifting their xy coordinates so that their centers align with the origin. During each forward iteration, we randomly sample 5120 points from one scene and apply data augmentation by randomly rotating the point cloud along each axis, adding a random translation sampled from a normal distribution with zero mean and unit variance, and applying a random scaling factor sampled from a uniform distribution in the range $[0.5, 1.5)$. Before passing a point cloud to the diffusion model, we re-scale it with a scale factor of 20 to ensure that most points lie within the range $[-1, 1]$. Finally, diffusion noise is added to the point cloud following the standard DDPM schedule [37].

6.2. SCR Mapping with Diffusion

During SCR mapping, the point clouds output by the Scene Coordinate Regression network are re-scaled using the same scale factor (20) as during training, before being fed into the diffusion model for noise estimation. If the batch size exceeds 5120, we randomly subsample the points down to 5120 to compute the diffusion regularization. This step is taken to prevent excessive processing time during point cloud encoding when the point cloud size is large. To balance the magnitude of diffusion regularization against the reprojection loss, we adopt the gradient normalization approach from DiffusionNeRF [99]. Specifically, the gradient of the regularization term is normalized with respect to itself and then scaled by a weight. In all experiments, this weight is set to 1000, with a warm-up phase spanning the first 1000 iterations after diffusion regularization begins at iteration 5000 of SCR mapping. During this warm-up, the weight increases linearly from 0 to 1000.

6.3. Prior Weights

For the depth distribution priors, we utilize a weight of $\lambda_{\text{reg}} = 0.1$. For the depth prior using RGB-D images, we use a weight of $\lambda_{\text{reg}} = 1$. These weights have been found by monitoring the magnitude of gradients stemming from the priors and the reprojection error during some mapping runs on ScanNet training sequences. For the weighting schema of the diffusion prior, see the previous section.

6.4. Diffusion-ACE Alignment

See Fig. 5 for a visualisation of the ACE mapping process versus a reverse diffusion process on the same scene. The beginning of ACE mapping does not align well with diffusion, hence we apply the diffusion prior only after a standby time of 5k iterations.

6.5. Point Cloud Visualization

Our point cloud visualizations were generated using the visualization code from PointFlow [100].

6.6. Details of Baseline Approaches

For ACE [10] and GLACE [93], we use their public code to reproduce their results on 7Scenes which have slight differences with the results reported in their original papers. Similarly, we obtain the results of ACE and GLACE on Indoor6 by running their code using the default settings.

7. Additional Results

7.1. Relocalization Results

7Scenes We conduct experiments with an alternative set of ground truth poses [9] stemming from an RGB-D SLAM system [82]. The relocalization performance is shown in Tab. 4. Analogously with the results obtained using SfM poses, presented in the main paper, our priors improve performance over the baseline models (ACE and GLACE) especially on the Stairs scene.

Indoor6 We report the per-scene relocalization results in Tab. 5 for ACE and GLACE with and without our diffusion prior. Results are averaged over 5 runs, except for EGFS [51], where results are taken from its paper. Additionally, we include model variants where '50K' denotes a batch size of 51200 points and 'dual' represents an ensemble of two models trained on a pre-clustering of the mapping camera poses [10]. For each variant, our model outperforms its respective baseline, demonstrating improved relocalization accuracy. Among SCR approaches, GLACE with our diffusion prior achieves the highest accuracy on this dataset.

7.2. Depth Evaluation on 7Scenes

We assess the point cloud quality after ACE mapping with and without our priors. To this end we compare depth values derived from the predicted scene coordinates with the ground truth depth images. In line with previous works [31, 50, 70], we apply standard metrics, including: Abs Rel, Sq Rel, RMSE, RMSE log, δ_1, δ_2 and δ_3. They are defined as follows:

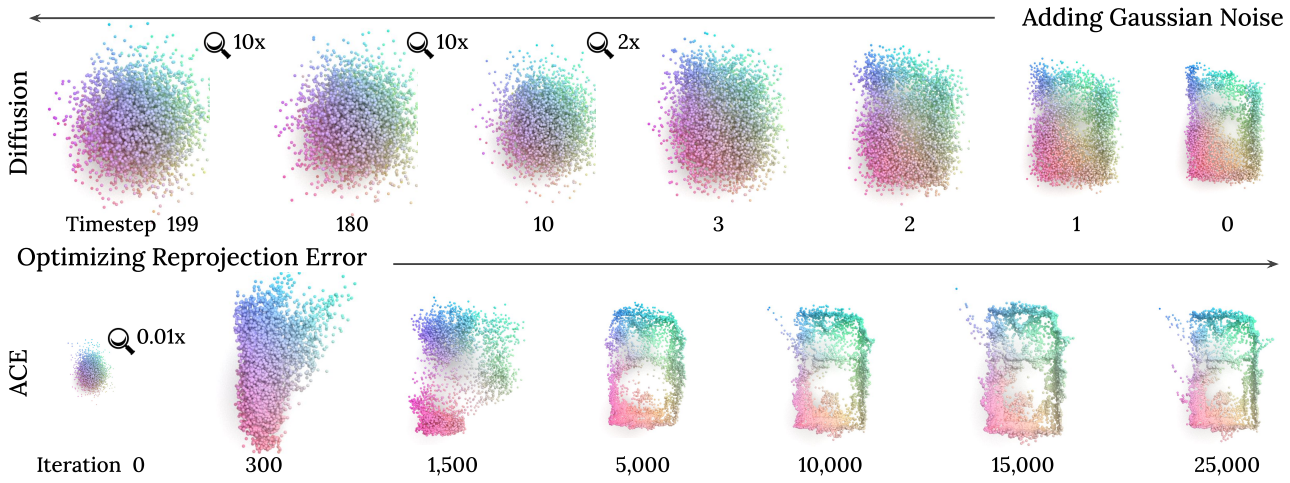


Figure 5. **Diffusion Process vs. ACE Training.** The evolution of point clouds during ACE training does not match the forward diffusion process over the full range. Hence, we align the diffusion time steps 10-0 with the ACE iterations 5,000-25,000.

Table 4. **Relocalization Results on 7Scenes with SLAM Poses.** We report the percentage of test images below a 5cm/5° pose error, mapping time and map size. Methods in "SCR w/ 3D" use depth or 3D point cloud supervision during mapping. Best results within the SCR groups are highlighted in **bold**.

Type	Method	Mapping Time	Map Size	Chess	Fire	Heads	Office	Pumpkin	Redkitchen	Stairs	Avg
FM	AS (SIFT) [77]	∼1.5h	∼200MB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	68.7%
	D.VLAD+R2D2 [38]	∼1.5h	∼1GB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	77.6%
	hLoc (SP+SG) [73, 74]	∼1.5h	∼2GB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	76.8%
	pixLoc [75]	∼1.5h	∼1GB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	75.7%
SCR w/ 3D	DSAC* [6]	15h	28MB	**97.3%**	94.0%	99.7%	**87.4%**	**62.9%**	**63.7%**	**83.4%**	**84.0%**
	SANet [101]	2.3min	550MB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	68.2%
	SRC [28]	2min	40MB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	55.2%
	ACE [10, 11] + DSAC* Loss [6]	5.5min	4MB	96.0%	94.0%	**99.9%**	84.5%	55.1%	57.8%	76.6%	80.6%
	ACE + Laplace NLL (Ours)	5.5min	4MB	97.1%	**94.8%**	99.8%	86.4%	57.5%	59.2%	82.9%	82.5%
SCR	DSAC* [6]	15h	28MB	95.3%	**94.5%**	98.1%	86.3%	61.6%	64.0%	67.6%	81.1%
	GLACE [93]	6min	9MB	98.5%	93.7%	99.7%	**90.2%**	61.9%	**73.9%**	54.0%	81.7%
	GLACE + Diffusion (Ours)	9min	9MB	**98.9%**	92.8%	99.3%	89.5%	**64.3%**	72.6%	56.4%	**82.0%**
	ACE [10]	5min	4MB	96.7%	92.4%	99.7%	86.0%	59.2%	60.3%	68.9%	80.5%
	ACE + Laplace NLL (Ours)	4.5min	4MB	96.6%	92.8%	99.8%	85.0%	57.8%	59.2%	72.4%	80.5%
	ACE + Laplace WD (Ours)	4.5min	4MB	96.7%	94.0%	99.6%	85.7%	59.7%	59.4%	71.0%	80.9%
	ACE + Diffusion (Ours)	8min	4MB	96.9%	93.8%	**99.8%**	85.7%	57.1%	59.8%	**74.3%**	81.1%

- Abs Rel: $\frac{1}{|\mathcal{V}|}\sum_{d\in\mathcal{V}}\|d-d_{\text{gt}}\|/d_{\text{gt}}$;
- Sq Rel: $\frac{1}{|\mathcal{V}|}\sum_{d\in\mathcal{V}}\|d-d_{\text{gt}}\|_2^2/d_{\text{gt}}$;
- RMSE: $\sqrt{\frac{1}{|\mathcal{V}|}\sum_{d\in\mathcal{V}}\|d-d_{\text{gt}}\|_2^2}$;
- RMSE log: $\sqrt{\frac{1}{|\mathcal{V}|}\sum_{d\in\mathcal{V}}\|\log d-\log d_{\text{gt}}\|_2^2}$;
- δ_i: % of $\mathcal{V}$ s.t. $\max(\frac{d}{d_{\text{gt}}},\frac{d_{\text{gt}}}{d})=\delta<i$;

where d is the estimated depth, d_{gt} is the ground truth depth, and $\mathcal{V}$ is the collection of all valid pixels on a depth map.

To generate per-frame depth maps we compute the scene coordinates for each mapping view, transform them into camera space, and derive the estimated depth from the z-coordinates. The average errors between these depth estimates and the ground truth across all scenes in 7Scenes are presented in Tab. 6. The results demonstrate that all our priors significantly reduce the depth error compared to the baseline approaches for both ACE and GLACE. In particular, the number of outlier points decreases as signified in the drastic reduction of the Sq Rel metric.

7.3. Analysis

Architecture for Point Cloud Encoder We compare the PVCNN architecture [53] with the Pointwise-Net used in [56]. Pointwise-Net processes each point independently, which limits its ability to capture structural relationships between points. As shown in Tab. 7 and Fig. 6, PVCNN outperforms Pointwise-Net in both relocalization and reconstruction tasks, showing the advantage when incorporating structural information.

Mask Threshold As described in the main paper, during SCR mapping, diffusion regularization is applied only

Table 5. **Per-Scene Relocalization Accuracy on Indoor6.** We report the percentage of test images below a 5cm, 5° pose error, mapping time and map size. Methods in "SCR w/ 3D" use the 3D point cloud as supervision during mapping. '50K' denotes a batch size of 51200 points and 'dual' represents an ensemble model with two clusters. Methods using our diffusion prior are denoted as *-Diff*.

Type	Method	Mapping Time	Map Size	scene1	scene2a	scene3	scene4a	scene5	scene6	Average
FM	hLoc (SP+SG)	∼3.3	∼1.5GB	70.5%	52.1%	86.0%	75.3%	58.0%	86.7%	71.4%
SCR (w/ 3D)	DSAC*	15h	28MB	18.7%	28.0%	19.7%	60.8%	10.6%	44.3%	30.4%
	SLD (300 LM)	5.5h	15MB	47.2%	48.2%	56.2%	67.7%	33.7%	52.0%	50.8%
	SLD (1000 LM)	44h	120MB	68.5%	62.6%	76.2%	77.2%	57.8%	78.0%	70.1%
SCR	DSAC*	15h	28MB	23.0%	33.9%	26.0%	67.1%	10.6%	50.2%	35.1%
	EGFS	21min	4.5MB	46.4%	60.6%	56.4%	78.7%	22.8%	71.6%	56.1%
	EGFS (dual)	21min	9MB	58.5%	59.1%	67.0%	76.1%	30.6%	75.9%	61.2%
	GLACE	11min	9MB	31.1%	44.8%	37.3%	**72.2%**	19.4%	60.1%	44.2% ± 1.8%
	GLACE-Diff	15min	9MB	**35.7%**	**46.5%**	**41.5%**	69.0%	**22.8%**	**62.7%**	**46.4%** ± 1.9%
	ACE	5min	4MB	24.5%	35.1%	34.4%	**58.9%**	15.7%	48.4%	36.2% ± 1.5%
	ACE-Diff	8min	4MB	**26.9%**	**35.3%**	**37.2%**	58.5%	**16.6%**	**50.3%**	**37.5%** ± 1.8%
	GLACE (50K)	33min	9MB	64.1%	**68.5%**	**73.1%**	**84.8%**	41.2%	85.2%	69.5% ± 1.4%
	GLACE-Diff (50K)	40min	9MB	**65.1%**	67.4%	73.0%	84.2%	**41.8%**	**85.9%**	**69.6%** ± 2.0%
	ACE (50K)	10min	4MB	**47.9%**	55.0%	60.5%	**77.3%**	26.4%	75.9%	57.2% ± 1.6%
	ACE-Diff (50K)	13min	4MB	**47.9%**	**56.0%**	**62.0%**	76.5%	**27.1%**	**77.7%**	**57.9%** ± 1.1%
	GLACE (dual)	22min	18MB	51.1%	56.8%	60.9%	75.9%	**30.5%**	75.6%	58.5% ± 1.9%
	GLACE-Diff (dual)	30min	18MB	**51.6%**	**58.2%**	**61.9%**	**76.7%**	28.2%	**76.5%**	**58.9%** ± 1.7%
	ACE (dual)	10min	8MB	41.4%	46.0%	54.4%	65.8%	20.5%	63.1%	48.5% ± 1.9%
	ACE-Diff (dual)	16min	8MB	**43.2%**	**48.7%**	**55.0%**	**66.1%**	**21.8%**	**64.4%**	**49.9%** ± 1.8%

Table 6. **Depth Evaluation on 7Scenes [82].** We report the errors between the estimated depth images, derived from the 3D scene points generated by SCR, against the ground truth depth images for each method.

	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
ACE [10]	0.48	24.25	2.26	0.33	0.87	0.93	0.96
ACE+Laplace NLL (Ours)	0.34	1.34	1.65	0.33	0.88	0.94	0.97
ACE+Laplace WD (Ours)	0.35	1.43	1.66	0.32	0.88	0.94	0.97
ACE+Diffusion (Ours)	0.32	1.17	1.62	0.32	0.88	0.95	0.97
GLACE [93]	0.40	25.57	2.36	0.25	0.89	0.95	0.97
GLACE+Diffusion (Ours)	**0.28**	**0.96**	**1.60**	**0.23**	**0.90**	**0.96**	**0.98**

to points with a reprojection error greater than 30 pixels. Tab. 8 compares the relocalization accuracy when regularization is applied to all points ($\kappa = 0$) and only to points with reprojection errors above 60 pixels ($\kappa = 60$). In the latter scenario, regularization affects fewer points, resulting in performance that closely resembles the original ACE. While applying diffusion to all points also improves upon the baseline ACE, the results indicate that $\kappa = 30$ achieves the best accuracy.

Frame Sample Rate Fig. 7 shows the impact of subsampling the number of mapping images in 7Scenes. Our diffusion prior mitigates the effect of scarce data.

Efficiency The diffusion regularization adds extra computation time to each ACE mapping iteration. To balance efficiency and accuracy, we reduce the frequency of applying the diffusion regularization by only implementing it every k mapping iterations. As shown in Tab. 9, setting $k = 4$ (i.e. running one diffusion iteration every 4 mapping steps) achieves the optimal trade-off, adding approximately 3 minutes to the total ACE mapping time.

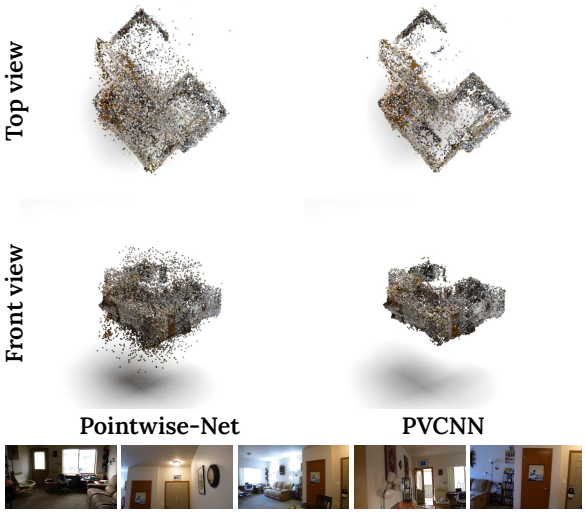


Figure 6. **Qualitative Results with Different Architectures for Point Cloud Encoder.** We compare the point clouds obtained after SCR mapping using diffusion regularization with different point cloud encoders. Pointwise-Net encodes every point independently whereas PCVNN captures structural information.

Table 7. **Ablation Study of the Diffusion Model Architecture.** We compare the relocalization accuracy (5cm, 5°) and depth errors on 7Scenes with different architectures of the diffusion point cloud encoder.

	Reloc Acc ↑	RMSE ↓	RMSE log ↓	δ_1 ↑
PVCNN [54]	**97.7%**	**1.62**	**0.32**	**0.883**
Pointwise-Net [56]	97.5%	1.64	0.69	0.879

Table 8. **Mask Threshold for Diffusion.** We compare the relocalization accuracy (5cm, 5°) on 7Scenes with different mask thresholds κ for applying diffusion regularisation.

	Chess	Fire	Heads	Office	Pumpkin	Redkitchen	Stairs	Avg
ACE	100.0%	99.5%	99.7%	100.0%	99.9%	98.6%	81.9%	97.1%
$\kappa = 0$	**100.0%**	99.4%	**100.0%**	99.8%	**99.3%**	98.4%	85.4%	97.5%
$\kappa = 30$	**100.0%**	**99.5%**	**100.0%**	**100.0%**	99.0%	**99.1%**	**86.2%**	**97.7%**
$\kappa = 60$	**100.0%**	99.6%	99.9%	99.8%	99.0%	98.3%	80.8%	96.8%

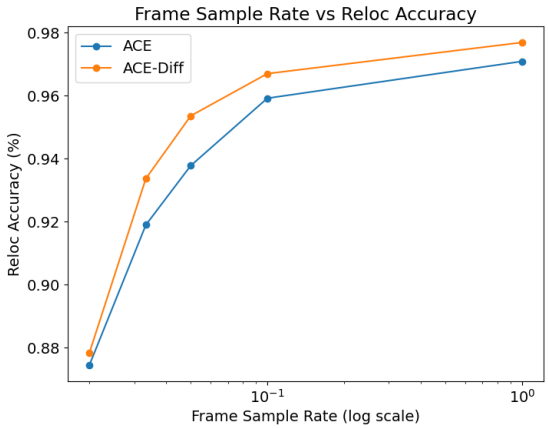


Figure 7. **Mapping Sample Rate vs. Accuracy on 7Scenes.** Subsampling the mapping frames leads to a decrease in relocalisation performance, but ACE with out diffusion prior (*ACE-Diff*) consistently outperforms ACE.

Table 9. **Efficiency-Accuracy Analysis for Regularization**. We analyze how the frequency of applying diffusion during mapping impacts both relocalization accuracy (5cm, 5°) and mapping time.

k	7Scenes		Indoor6 (N=51200)	
	Reloc Acc ↑	Time	Reloc Acc ↑	Time
1	97.7%	18min	57.5%	24min
2	97.7%	11min	57.5%	17min
4	97.7%	8min	57.9%	13min
8	97.6%	6min	57.2%	12min

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Scene Coordinate Reconstruction Priors
Publication Status	□Published ☑Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	**Bian, W.**, Barroso-Laguna, A., Cavallari, T., Prisacariu, V. A. & Brachmann E. (2025). In Proceedings of the IEEE/CVF International Conference on Computer Vision.

Student Confirmation

Student Name:	Wenjing Bian		
Contribution to the Paper	• Joint conception and sole validation of the idea • Design of the architecture and framework • Implementation and running of most of the experiments • Research of prior work • Writing of the paper		
Signature	Wenjing Bian	Date	April 22, 2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Victor Adrian Prisacariu		
Supervisor comments The description is accurate		
Signature VPA	Date	April 22, 2025

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 8

Conclusion

In this thesis, we investigate a range of learning-based priors designed to enhance the performance of 3D vision tasks at both the object and scene level. These priors help to resolve ambiguities in 3D geometry that cannot be inferred solely from visual cues within the images, enabling models to develop a more robust understanding of underlying structures. Consequently, they contribute to improved performance in downstream tasks such as 3D reconstruction, 3D detection, novel view synthesis and relocalisation.

We begin by addressing object-level 3D vision tasks, where the input is a single RGB image. In Chapter 3, we present a method for monocular object reconstruction that learns ray-based priors from the input image, capturing both categorical information and high-level photometric cues. Combined with a ray-based representation, this approach enhances reconstruction accuracy, improves inference efficiency, and demonstrates strong generalisation to unseen object categories and real-world images.

In Chapter 4, we introduce a category-agnostic 3D object detection model from a single image, which leverages priors including image features, 2D bounding boxes, and depth information. By applying diffusion to randomly initialised 3D

boxes, we eliminate the reliance on regional proposals. The model achieves strong performance on in-domain tasks and demonstrates impressive generalisation to images captured in the wild.

We then shift our focus to scene-level 3D vision, where we leverage learning-based priors to infer 3D scenes from multi-view images. In Chapter 5, we present an approach for optimising NeRF with unknown camera poses, incorporating geometric priors derived from an off-the-shelf monocular depth estimation model. The depth prior helps reduce NeRF's inherent ambiguities and provides constraints on pairwise camera poses. Consequently, this approach improves NeRF reconstruction quality, enhances view synthesis, and increases pose estimation accuracy.

In Chapter 6, we introduce the Pose Residual Field (PoRF) to NeuS for camera pose refinement. Compared to optimising individual pose parameters independently, PoRF imposes additional implicit constraints across the entire trajectory. Combined with an epipolar geometry loss, this approach achieves improved performance in both pose accuracy and reconstruction quality.

Finally, in Chapter 7, we introduce geometric priors to regularise scene coordinates during training. These priors are derived from depth distributions and a diffusion model that captures high-level indoor geometries. By incorporating them into state-of-the-art SCR models, we achieve more accurate scene geometry and improved performance on downstream tasks.

8.1 Future Work

Recent advances in 3D vision have led to the development of increasingly large-scale models for 3D tasks [178, 179, 180, 14]. However, their generalisation remains limited by the scarcity and lack of diversity in available datasets, high computational demands, and the inherent complexity of handling multiple 3D representa-

tions.

On the other hand, large foundation models in language [181, 182, 183], 2D vision [8, 11, 184, 18] and multimodal learning [185, 186, 187] have achieved wide applicability across domains. While not explicitly trained for 3D understanding, these models can still provide useful priors for 3D vision tasks. For example, features from pre-trained 2D diffusion models such as Stable Diffusion [116] and ControlNet [18] have demonstrated emergent capabilities such as 3D awareness [188] and correspondence estimation [189], enhancing performance in downstream applications.

Nevertheless, such priors often lack precise 3D geometric grounding and may be unreliable for geometry-intensive tasks. Developing unified, generalisable priors tailored to 3D vision remains an important goal. A promising direction is to integrate these learned priors with 3D-specific knowledge and traditional, non-learning-based constraints—such as multi-view consistency and temporal coherence—to build more robust and versatile 3D vision systems.

In parallel, optimising model architectures and pipelines to improve training and inference efficiency is a practical and impactful goal for 3D vision. Reducing computational costs not only makes large-scale 3D models more accessible but also enables on-device deployment, unlocking a wider range of real-world applications and user scenarios.

Bibliography

[1] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[2] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[3] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial Intelligence*, 1981.

[4] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 2002.

[5] James M Coughlan and Alan L Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *ICCV*, 1999.

[6] David C Lee, Martial Hebert, and Takeo Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, 2009.

[7] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.

[8] 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 *ICCV*, 2023.

[9] Rejin Varghese and M Sambath. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In *ADICS*, 2024.

[10] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[11] 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. *Transactions on Machine Learning Research (TMLR)*, 2023.

[12] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[14] 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 *CVPR*, 2024.

[15] Kai Gu, Thomas Maugey, Sebastian Knorr, and Christine Guillemot. Omni-nerf: neural radiance field from 360 image captures. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, 2022.

[16] Zhuoling Li, Xiaogang Xu, SerNam Lim, and Hengshuang Zhao. Unimode: Unified monocular 3d object detection. In *CVPR*, 2024.

[17] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021.

[18] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

[19] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *Entropy*, 2023.

[20] Ruiqi Gao*, Aleksander Holynski*, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul P. Srinivasan, Jonathan T. Barron, and Ben Poole*. Cat3d: Create anything in 3d with multi-view diffusion models. In *NeurIPs*, 2024.

[21] Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T Freeman, and Jiajun Wu. Wonderworld: Interactive 3d scene generation from a single image. *arXiv preprint arXiv:2406.09394*, 2024.

[22] Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T Barron, and Aleksander Holynski. Cat4d: Create anything in 4d with multi-view video diffusion models. *arXiv preprint arXiv:2411.18613*, 2024.

[23] Haiyu Zhang, Xinyuan Chen, Yaohui Wang, Xihui Liu, Yunhong Wang, and Yu Qiao. 4diffusion: Multi-view video diffusion model for 4d generation. *NeurIPS*, 2025.

[24] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016.

[25] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019.

[26] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *ICCV*, 2019.

[27] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. *NeurIPs*, 2019.

[28] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *CVPR*, 2019.

[29] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *CVPR*, 2020.

[30] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3D: A large benchmark and model for 3D object detection in the wild. In *CVPR*, 2023.

[31] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *CoRL*, 2022.

[32] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *CVPR*, 2020.

[33] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In *WACV*, 2022.

[34] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[35] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 2023.

[36] 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.

[37] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021.

[38] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *ICCV*, 2021.

[39] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *CVPR*, 2013.

[40] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *ECCV*, 2014.

[41] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. In *ECCV*, 2024.

[42] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *CVPR*, 2017.

[43] Eric Brachmann and Carsten Rother. Visual camera re-localization from RGB and RGB-D images using DSAC. *PAMI*, 2021.

[44] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *CVPR*, 2023.

[45] Fangjinhua Wang, Xudong Jiang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. Glace: Global local accelerated coordinate encoding. In *CVPR*, 2024.

[46] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M Seitz, and Richard Szeliski. Building rome in a day. In *ICCV*, 2009.

[47] Changchang Wu. Towards linear-time incremental structure from motion. In *3DV*, 2013.

[48] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *TPAMI*, 2007.

[49] Georg Klein and David Murray. Parallel tracking and mapping on a camera phone. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2009.

[50] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[51] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *ICRA*, 2014.

[52] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017.

[53] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017.

[54] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *NeurIPS*, 2019.

[55] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021.

[56] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. In *CVPR*, 2003.

[57] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *NeurIPs*, 2014.

[58] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *TPAMI*, 2020.

[59] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018.

[60] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *ICCV*, 2023.

[61] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 1987.

[62] Akio Doi and Akio Koide. An efficient method of triangulating equi-valued surfaces by using tetrahedral cells. *IEICE TRANSACTIONS on Information and Systems*, 1991.

[63] John C Hart. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 1996.

[64] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *NeurIPS*, 2021.

[65] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *NeurIPS*, 2021.

[66] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *CVPR*, 2021.

[67] 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.

[68] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 2022.

[69] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021.

[70] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, 2022.

[71] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *CVPR*, 2023.

[72] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021.

[73] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021.

[74] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *ECCV*, 2022.

[75] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *CVPR*, 2023.

[76] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *CVPR*, 2023.

[77] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, 2021.

[78] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021.

[79] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

[80] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *CVPR*, 2022.

[81] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *CVPR*, 2022.

[82] Jamie Wynn and Daniyar Turmukhambetov. DiffusioNeRF: Regularizing Neural Radiance Fields with Denoising Diffusion Models. In *CVPR*, 2023.

[83] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *CVPR*, 2022.

[84] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *ICCV*, 2021.

[85] Kyle Gao, Yina Gao, Hongjie He, Denning Lu, Linlin Xu, and Jonathan Li. Nerf: Neural radiance field in 3d vision, a comprehensive review. *arXiv preprint arXiv:2210.00379*, 2022.

[86] Tommaso Cavallari, Stuart Golodetz, Nicholas A Lord, Julien Valentin, Luigi Di Stefano, and Philip HS Torr. On-the-fly adaptation of regression forests for online camera relocalisation. In *CVPR*, 2017.

[87] Tommaso Cavallari, Stuart Golodetz, Nicholas A Lord, Julien Valentin, Victor A Prisacariu, Luigi Di Stefano, and Philip HS Torr. Real-time rgb-d camera pose estimation in novel scenes using a relocalisation cascade. *TPAMI*, 2019.

[88] Tommaso Cavallari, Luca Bertinetto, Jishnu Mukhoti, Philip Torr, and Stuart Golodetz. Let's take this online: Adapting scene coordinate regression network predictions for online rgb-d camera relocalisation. In *3DV*, 2019.

[89] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *CVPR*, 2018.

[90] Eric Brachmann, Frank Michel, Alexander Krull, Michael Y. Yang, Stefan Gumhold, and Carsten Rother. Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image. In *CVPR*, 2016.

[91] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *ICCV*, 2019.

[92] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *CVPR*, 2020.

[93] Tommaso Cavallari, Stuart Golodetz, Nicholas A. Lord, Julien Valentin, Victor A. Prisacariu, Luigi Di Stefano, and Philip H. S. Torr. Real-time rgb-d camera pose estimation in novel scenes using a relocalisation cascade. *PAMI*, 2019.

[94] Eric Brachmann and Carsten Rother. Neural- Guided RANSAC: Learning where to sample model hypotheses. In *ICCV*, 2019.

[95] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *TPAMI*, 2003.

[96] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.

[97] Stefan Popov, Pablo Bauszat, and Vittorio Ferrari. Corenet: Coherent 3d scene reconstruction from a single rgb image. In *ECCV*, 2020.

[98] Wenjing Bian, Zirui Wang, Kejie Li, and Victor Adrian Prisacariu. Ray-onet: Efficient 3d reconstruction from a single rgb image. In *BMVC*, 2021.

[99] Zhizhong Han, Guanhui Qiao, Yu-Shen Liu, and Matthias Zwicker. Seqxy2seqz: Structure learning for 3d shapes by sequentially predicting 1d occupancy segments from 2d coordinates. In *ECCV*, 2020.

[100] Omid Poursaeed, Matthew Fisher, Noam Aigerman, and Vladimir G Kim. Coupling explicit and implicit surface representations for generative 3d modeling. In *ECCV*, 2020.

[101] Kejie Li, Yansong Tang, Victor Adrian Prisacariu, and Philip HS Torr. Bnv-fusion: Dense 3d reconstruction using bi-level neural volume fusion. In *CVPR*, 2022.

[102] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *CVPR*, 2022.

[103] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading: a survey. *TPAMI*, 1999.

[104] Jean-Denis Durou, Maurizio Falcone, and Manuela Sagona. Numerical methods for shape-from-shading: A new survey with benchmarks. *Computer Vision and Image Understanding*, 2008.

[105] Aldo Laurentini. The visual hull concept for silhouette-based image understanding. *TPAMI*, 1994.

[106] Bruce Guenther Baumgart. *Geometric modeling for computer vision.* Stanford University, 1974.

[107] Michael Daum and Gregory Dudek. On 3-d surface reconstruction using shape from shadows. In *CVPR*, 1998.

[108] Silvio Savarese, Marco Andreetto, Holly Rushmeier, Fausto Bernardini, and Pietro Perona. 3d reconstruction by shadow carving: Theory and practical evaluation. *International journal of computer vision*, 2007.

[109] John Aloimonos. Shape from texture. *Biological cybernetics*, 1988.

[110] Boaz J. Super and Alan C. Bovik. Shape from texture using local spectral moments. *TPAMI*, 1995.

[111] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *TPAMI*, 2008.

[112] Derek Hoiem, Alexei A Efros, and Martial Hebert. Automatic photo pop-up. In *ACM SIGGRAPH*. 2005.

[113] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, 2017.

[114] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, 2018.

[115] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *NeurIPS*, 2019.

[116] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. 2022 ieee. In *CVPR*, 2021.

[117] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. In *ICCV*, 2023.

[118] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.

[119] Olivier Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT press, 1993.

[120] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu*, 2000.

[121] Sameer Agarwal, Noah Snavely, Steven M Seitz, and Richard Szeliski. Bundle adjustment in the large. In *ECCV*, 2010.

[122] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006.

[123] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016.

[124] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, 2018.

[125] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *CVPR*, 2018.

[126] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *ICCV*, 2017.

[127] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. In *CVPR*, 2023.

[128] Yu Chen and Gim Hee Lee. Dbarf: Deep bundle-adjusting generalizable neural radiance fields. In *CVPR*, 2023.

[129] Axel Levy, Mark Matthews, Matan Sela, Gordon Wetzstein, and Dmitry Lagun. Melon: Nerf with unposed images in so (3). In *3DV*, 2024.

[130] Eric Brachmann and Carsten Rother. Learning Less is More - 6D Camera Localization via 3D Surface Regression. In *CVPR*, 2018.

[131] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPs*, 2020.

[132] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

[133] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2020.

[134] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

[135] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[136] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023.

[137] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *ICCV*, 2023.

[138] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[139] Julia Wolleb, Robin Sandkühler, Florentin Bieder, Philippe Valmaggia, and Philippe C Cattin. Diffusion models for implicit image segmentation ensembles. In *MIDL*, 2022.

[140] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126*, 2021.

[141] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes Lutz Schönberger. Global structure-from-motion revisited. In *ECCV*, 2024.

[142] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *ICCV*, 2023.

[143] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*, 2024.

[144] Yiquan Duan, Xianda Guo, and Zheng Zhu. Diffusiondepth: Diffusion denoising approach for monocular depth estimation. In *ECCV*, 2024.

[145] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2022.

[146] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*, 2023.

[147] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *ICCV*, 2021.

[148] Jaehyeok Shim, Changwoo Kang, and Kyungdon Joo. Diffusion-based signed distance fields for 3d shape generation. In *CVPR*, 2023.

[149] Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulo, Peter Kontschieder, and Matthias Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. In *CVPR*, 2023.

[150] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Viewset diffusion:(0-) image-conditioned 3d generative models from 2d data. In *ICCV*, 2023.

[151] Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Realfusion: 360deg reconstruction of any object from a single image. In *CVPR*, 2023.

[152] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

[153] Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Diffusion-sdf: Text-to-shape via voxelized diffusion. In *CVPR*, 2023.

[154] Jingyu Zhuang, Chen Wang, Liang Lin, Lingjie Liu, and Guanbin Li. Dreameditor: Text-driven 3d scene editing with neural fields. In *SIGGRAPH Asia*, 2023.

[155] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *CVPR*, 2024.

[156] Se-Ho Kim, Inyong Koo, Inyoung Lee, Byeongjun Park, and Changick Kim. Diffref3d: A diffusion-based proposal refinement framework for 3d object detection. *arXiv preprint arXiv:2310.16349*, 2023.

[157] Yasiru Ranasinghe, Deepti Hegde, and Vishal M Patel. Monodiff: Monocular 3d object detection and pose estimation with diffusion models. In *CVPR*, 2024.

[158] Xiangyu Chen, Zhenzhen Liu, Katie Z Luo, Siddhartha Datta, Adhitya Polavaram, Yan Wang, Yurong You, Boyi Li, Marco Pavone, Wei-Lun Chao, et al. Diffubox: Refining 3d object detection with point diffusion. *arXiv preprint arXiv:2405.16034*, 2024.

[159] Zhen Wang, Dongyuan Li, and Renhe Jiang. Diffusion models in 3d vision: A survey. *arXiv preprint arXiv:2410.04738*, 2024.

[160] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3d lidar using fully convolutional network. *arXiv preprint arXiv:1608.07916*, 2016.

[161] Jorge Beltrán, Carlos Guindel, Francisco Miguel Moreno, Daniel Cruzado, Fernando Garcia, and Arturo De La Escalera. Birdnet: a 3d object detection framework from lidar information. In *ITSC*, 2018.

[162] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *CVPR*, 2017.

[163] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *IROS*, 2018.

[164] Xin Zhou, Jinghua Hou, Tingting Yao, Dingkang Liang, Zhe Liu, Zhikang Zou, Xiaoqing Ye, Jianwei Cheng, and Xiang Bai. Diffusion-based 3d object detection with random boxes. In *PRCV*, 2023.

[165] Wenjing Bian, Zirui Wang, and Andrea Vedaldi. Catfree3d: Category-agnostic 3d object detection with diffusion. In *3DV*, 2025.

[166] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 2018.

[167] Xiao Zheng, Xiaoshui Huang, Guofeng Mei, Yuenan Hou, Zhaoyang Lyu, Bo Dai, Wanli Ouyang, and Yongshun Gong. Point Cloud Pre-Training with Diffusion Models . In *CVPR*, 2024.

[168] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. In *CVPR*, 2023.

[169] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023.

[170] Gene Chou, Yuval Bahat, and Felix Heide. Diffusion-sdf: Conditional generative modeling of signed distance functions. In *ICCV*, 2023.

[171] Junsheng Zhou, Weiqi Zhang, and Yu-Shen Liu. Diffgs: Functional gaussian splatting diffusion. *NeurIPS*, 2025.

[172] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019.

[173] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[174] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv*, 2022.

[175] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *ECCV*, 2020.

[176] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019.

[177] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[178] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024.

[179] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r. In *ECCV*, 2024.

[180] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *ICCV*, 2023.

[181] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

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

[183] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[184] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021.

[185] 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 *ICML*, 2021.

[186] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 2023.

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

[188] Chenfeng Xu, Huan Ling, Sanja Fidler, and Or Litany. 3difftection: 3d object detection with geometry-aware diffusion features. In *CVPR*, 2024.

[189] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *NeurIPs*, 2023.