Deep Geometry-Prior for Absolute Pose Regression

Shuai Chen
Linacre College
University of Oxford

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

To my family, for their unwavering support,
and to my daughter, Ling'Er, no boyfriend until 2040.

Abstract

Camera relocalisation, a core problem in computer vision and robotics, is crucial for transformative technologies such as augmented reality, autonomous navigation, and 3D scene reconstruction. This thesis seeks to push the boundaries of visual relocalisation by studying various methods of integrating geometric priors into the end-to-end deep learning models, addressing three key aspects: training, inference, and network architecture. By harnessing innovative methods, including Neural Radiance Fields and 3D Gaussian Splatting, alongside map-relative pose regression, the thesis contributes to more precise, efficient, and scalable solutions for camera relocalisation.

The research introduces novel training paradigms that incorporate implicit 3D-based direct photometric and feature-metric matching to absolute pose regression (APR) models. Techniques such as Direct-PoseNet and DFNet enhance APR performance through differentiable rendering.

The investigation then shifts to integrating 3D geometry at inference time, proposing advanced post-processing methods like neural feature synthesis-based pose refinement, uncertainty-aware hierarchical pose refinement, and efficient pose refinement using 3D Gaussians and 3D foundation models. These frameworks demonstrate significant improvements in pose estimation accuracy across various benchmarks.

Finally, the thesis introduces a geometrically informed network architectural design for APR, a map-relative pose regression framework that bridges the gap between end-to-end deep networks with 3D structure-based methods, enabling scalable and robust relocalisation in dynamic and unvisited environments.

Keywords – Camera Relocalisation, Absolute Pose Regression, Scene Coordinate Regression, Direct Matching, NeRF, 3D Gaussians Splatting, Novel View Synthesis, Semi-supervised Learning.

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.

Shuai Chen, March 2025.

Acknowledgement

Firstly, I am incredibly grateful to my supervisor, Professor Victor Adrian Prisacariu, for his unwavering support and invaluable guidance throughout my DPhil study. His expertise and insightful advice have been instrumental in shaping my growth as a researcher. Beyond his professional guidance, Prof. Prisacariu's humor and optimistic outlook have been a constant source of motivation and encouragement during the most challenging times of my studies. His ability to approach problems with both intellectual rigor and a cheerful perspective has taught me to maintain a balanced and positive mindset, both in my work and in life.

I would like to extend my sincere thanks to my labmates, collaborators, and friends who have been an integral part of my DPhil journey. Their kindness and support have made my time at Oxford truly memorable. I have cherished every moment spent working, learning, and growing alongside them, and I am deeply grateful for the encouragement and laughter we shared.

A special thank you to Zirui Wang, Xinghui Li, Wenjing Bian, Michael Hobley, Theo Costain, Marcelo Gennari do Nascimento, Henry Howard-Jenkins, Kejie Li, Jia-Wang Bian, Lixiong Chen, Gratianus Wesley Putra Data, Yash Bhalgat, Jing Wu, Brandon Smart, and Xianzheng Ma. Together, we have navigated the challenges of research, celebrated milestones, and created countless memories that I will treasure for years to come. Whether through late-night brainstorming sessions, collaborative problem-solving, or simply enjoying life outside the lab, you all have made these past four years an extraordinary experience.

Lastly, I express my deepest gratitude to my family. Your unconditional love and encouragement have been my greatest source of strength and inspiration, allowing me to pursue my dreams and overcome every challenge along the way. To my wife ZZ Li and my daughter Ling'Er, I will be forever grateful for your presence in my life, which reminds me every day of what truly matters. To my entire family, thank you for your unconditional love and for always being there for me. This accomplishment is as much yours as it is mine, and I dedicate it to you with all my heart.

Contents

Chapter 1

Introduction

Camera relocalisation is a fundamental problem in computer vision and robotics, with widespread applications in augmented reality (AR), virtual reality (VR), autonomous navigation, and 3D scene reconstruction. It involves estimating the camera's six degrees of freedom (6-DOF) pose relative to a known scene, which includes both its position and orientation. Accurate pose estimation is essential for modern intelligent systems to understand and interact with the 3D world. In AR and VR applications, it anchors virtual content to the physical world in real time. In autonomous robotic systems, it facilitates precise coordination between spatial perception and action. In 3D reconstruction, it provides the spatial grounding necessary for building coherent models of the environment.

Across these diverse applications, camera relocalisation indeed serves as the backbone for spatial understanding, and its performance directly impacts the reliability of downstream tasks. To solve this problem, various algorithms and learning-based methods have been proposed, each with different trade-offs in terms of accuracy, efficiency, and scalability. On one end of the spectrum lies the traditional approach that typically relies on establishing 2D-3D correspondences and solving the pose using Perspective-n-Point (PnP) solvers [52, 57]. While effective,

these methods are relatively computationally expensive and often challenging to deploy on resource-constrained devices.

At the other end of the spectrum, deep learning advances have introduced Absolute Pose Regression (APR) techniques [79, 157], where neural networks predict camera poses directly from RGB images. Despite offering advantages in efficiency and deployment, APR methods often struggle with accuracy and generalisability. A potential reason for this, as shown in previous studies [144, 15, 78], is that APR networks lack explicit mechanisms for 3D geometric reasoning in their formulation.

This thesis focuses on closing the gap between deep-learning-based APR and state-of-the-art camera relocalisation methods, by enabling APRs to better model and leverage 3D scene structure. To this end, it introduces novel methods for improving the training, inference, and architecture of APR-based camera relocalisation, leveraging emerging 3D scene representation techniques such as Neural Radiance Fields (NeRF) [111, 56], 3D Gaussian Splatting (3D-GS) [80, 25, 51], and dense scene geometry networks [155, 12, 11].

The first part[1] introduces improvements in APR's training strategies by incorporating 3D supervision beyond the traditional 2D training pipeline. Specifically, Chapter 3 and Chapter 4 propose two novel training schemes to integrate photometric and feature-metric direct matching into the APR training process. Furthermore, these methods facilitate complementary training with unlabelled data, which is particularly useful in real-world environments where ground-truth data is scarce or expensive to obtain. These improvements enhance the accuracy of APR without increasing computational complexity during inference.

The second part aims to improve accuracy via changes to the inference process. Chapter 5 through Chapter 7 propose novel test-time refinement strategies that

[1]To present our research within a coherent methodological framework, the main chapters of this thesis are organised into four parts: Part I, focusing on the training of Absolute Pose Regression; Part II, covering its inference process; Part III, addressing its architectural design; and Part IV, summarising existing achievements and discussing future work directions.

yield results comparable to, and in some cases even surpassing, state-of-the-art (SOTA) visual relocalisation methods. Chapter 5 presents a test-time refinement pipeline leveraging customised neural feature fields. Chapter 6 introduces a hierarchical refinement strategy that optimises computational efficiency for APR test-time refinement. Finally, Chapter 7 details an advanced camera pose refinement method that enhances both APR-based and non-APR relocalisation methods by utilising 3D Gaussian-based representations and 3D foundation models.[2]

The third part proposes a scheme to integrate 3D geometry directly into APR network architecture. To the best of our knowledge, this is the first work to explicitly incorporate 3D geometric principles into APR network design. Moreover, state-of-the-art APR methods require prolonged training times, ranging from tens of hours to several days, limiting their adaptability to new environments. Chapter 8 introduces a novel APR framework that combines the advantages of 3D geometry-based and end-to-end APR approaches, empowering APR with explicit 3D geometric reasoning. As a result, this framework enables deployment in unseen environments within minutes of mapping time while maintaining real-time efficiency and end-to-end simplicity.

The research trajectory presented in this thesis emerged from a systematic exploration of the accuracy-scalability-robustness trade-offs inherent in camera relocalisation. As the field of APR evolved from isolated pose regression toward integrating dense geometric representations, each stage of this work addressed specific limitations identified in its predecessor. This progression transitioned from enhancing APR training with implicit 3D priors to developing efficient inference-time refinement frameworks leveraging NeRF and 3D Gaussian Splatting, and

[2]Throughout the development of Part I and Part II, our underlying scene representations evolved from neural radiance fields to neural feature fields, and ultimately to 3D Gaussian Splatting, improving rendering quality but remaining sensitive to scene scale, model capacity, and initial pose accuracy. Detailed discussions on these constraints in challenging scenarios (e.g., city-scale, seasonal changes) are provided in Chapter 7 and Chapter 9.

ultimately culminated in the design of geometry-aware network architectures that enable fast mapping to novel scenes. This methodological evolution reflects an overarching strategy to bridge the gap between the high precision of geometric methods and the end-to-end efficiency required for real-world deployment.

The remainder of this thesis is organised as follows: Section 1.1 lists the key contributions of our research to the field of camera relocalisation, categorised into geometric priors in training, inference, and network architecture. Section 1.2 provides a thesis outline, and Section 1.3 presents a comprehensive list of related publications.

1.1 Key Contributions

This thesis contributes to the field of camera relocalisation by introducing novel methodologies that advance state-of-the-art APR techniques. It addresses key challenges in facilitating the use of geometric priors in training, inference, and network architecture, proposing solutions that improve accuracy, efficiency, and generalisability. Specifically, 3D geometric priors[3] encapsulate fundamental spatial constraints and relationships inherent in three-dimensional environments, such as depth consistency and scene rigidity, which have potentials to enhance an APR network's ability to model real-world structures. This thesis demonstrates that integrating 3D geometric priors improves the performance of APR, effectively addressing key weaknesses of existing APR approaches. The following sections list each contribution with respect to these challenges.

[3]In this thesis, we refer 3D geometric priors as representations of 3D scene structure (explicit or implicit) that assists or regularises pose prediction.

1.1.1 Geometric Priors in Training

While APR methods offer computational efficiency, they often overlook 3D geometric principles in their feed-forward formulation and struggle to generalise beyond training data [144]. Aiming to address this issue, Chapter 3 presents Direct-PoseNet, which combines an APR network with a differentiable novel view synthesiser based on volumetric rendering, achieving improved accuracy through photometric supervision. Additionally, this framework enables the utilisation of unlabeled images to further refine camera pose estimation.

Building upon this foundation, Chapter 4 introduces DFNet, which replaces the photometric formulation with a robust direct feature-metric matching approach. This design mitigates challenges posed by photometric inconsistencies, which are frequently seen in visual relocalisation scenarios, and thereby improves pose regression performance.

In summary, Chapter 3 and Chapter 4 propose two training frameworks that incorporate direct photometric and feature-metric supervision, advancing the robustness and accuracy of APR-based relocalisation methods.

1.1.2 Geometric Priors in Inference

State-of-the-art visual relocalisation methods often employ geometric refinement during test time to enhance accuracy. However, end-to-end approaches such as APR traditionally lack mechanisms to incorporate 3D geometry at inference, limiting their ability to refine poses dynamically. Existing APR efforts that explored test-time refinement often show limited effectiveness due to reliance on coarse constraints or suboptimal optimisation strategies [115, 15, 154].

To address these challenges, this thesis introduces a novel inference pipeline designed to integrate APR with implicit geometric constraints during test time.

Chapter 5 details the design and implementation of a neural feature field named NeFeS. We demonstrate that NeFeS enables substantial performance improvements for any existing APR method at test time, bridging the gap between APR efficiency and the geometric reasoning capabilities of traditional methods.

While test-time refinement is a promising avenue for improving APR accuracy, it inevitably introduces additional computational overhead. To mitigate this, we propose two solutions. The first, detailed in Chapter 6, leverages hierarchical refinement through uncertainty estimation. Observing that not all estimated poses have equal reliability, we develop an implicit database to store prior knowledge and precompute confidence estimates for pose regression, thereby optimising the refinement process.

The second approach, presented in Chapter 7, extends the NeFeS framework by introducing GS-CPR, a novel pose refinement method based on 3D Gaussian Splatting [80] and the 3D foundation model Mast3R [88]. This design enhances efficiency and scalability, achieving state-of-the-art results across multiple benchmarks compared to both APR-based and geometry-based relocalisation methods.

In summary, Chapter 5 details the development of NeFeS, offering a solution that integrates implicit geometric reasoning into the APR's testing framework. Chapter 6 addresses the computational overhead of test-time refinement by introducing a hierarchical approach with uncertainty estimation. Chapter 7 further enhances the efficiency and scalability of NeFeS through 3D-GS and 3D foundation models.

1.1.3 Geometric Priors in Network Architecture

Our work on applying geometric priors with test-time refinement has demonstrated substantial improvements in pose estimation accuracy. Nevertheless, state-of-the-art APR methods require vast amounts of training data and prolonged training

times, often ranging from tens of hours to several days. Moreover, the entire training process must be repeated for every new environment because the traditional APR model is trained specifically for a single scene. These factors make APR highly inefficient for large-scale deployment, particularly in dynamic or previously unseen environments where adaptability is crucial.

To address these challenges, Chapter 8 introduces map-relative pose regression (*marepo*), a framework that builds on the principles of APR but integrates scene-specific geometric priors. By leveraging a transformer-based pose regressor trained across hundreds of scenes, *marepo* learns a generic mapping between scene-specific representations and camera poses. As a result, the pose regressor can be applied to new map representations immediately or after mere minutes of fine-tuning, making it one of the most scalable APR frameworks available today.

1.2 Thesis Outline

This thesis is organised in an integrated thesis style[4], in which the bulk of some chapters (Chapter 3 — Chapter 8) are presented in the format as when they were originally published.

The outline of the thesis is as follows. Chapter 2 provides a literature review of relevant works. Chapter 3 to Chapter 8 present our main research outcomes, each corresponding to an accepted publication as mentioned in Section 1.1. Finally, Chapter 9 concludes the research achievements presented in this thesis and discusses the future direction of the work.

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

1.3 List of Publications

Chapter 3: "Direct-PoseNet: Absolute Pose Regression with Photometric Consistency". **Shuai Chen**, Zirui Wang, Victor Adrian Prisacariu. In *International Conference on 3D Vision (3DV)*, 2021.

Chapter 4: "DFNet: Enhance Absolute Pose Regression with Direct Feature Matching". **Shuai Chen**, Xinghui Li, Zirui Wang, Victor Adrian Prisacariu. In *European Conference on Computer Vision (ECCV)*, 2022.

Chapter 5: "Neural Refinement for Absolute Pose Regression with Feature Synthesis". **Shuai Chen**, Yash Bhalgat, Xinghui Li, Jiawang Bian, Kejie Li, Zirui Wang, Victor Adrian Prisacariu. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Chapter 6: "HR-APR: APR-agnostic Framework with Uncertainty Estimation and Hierarchical Refinement for Camera Relocalisation". Changkun Liu, **Shuai Chen**, Yukun Zhao, Huajian Huang, Victor Adrian Prisacariu, Tristan Braud. In *International Conference on Robotics and Automation (ICRA)*, 2024.

Chapter 7: "GS-CPR: Efficient Camera Pose Refinement via 3D Gaussian Splatting". Changkun Liu, **Shuai Chen**, Yash Bhalgat, Siyan Hu, Zirui Wang, Ming Cheng, Victor Adrian Prisacariu, Tristan Braud. In *International Conference on Learning Representations (ICLR)*, 2025.

Chapter 8: "Map-Relative Pose Regression for Visual Re-Localization". **Shuai Chen**, Tommaso Cavallari, Victor Adrian Prisacariu, Eric Brachmann. In *Conference on Computer Vision and Pattern Recognition (CVPR Highlight)*, 2024.

Publications Not Included

During my DPhil I also contributed to the following publications. We do not include them since they are only loosely connected to this thesis.

⇒ **"Deep Online Video Stabilization using IMU Sensors".** Chen Li, Li Song, **Shuai Chen**, Rong Xie, Wenjun Zhang. In *IEEE Transactions on Multimedia (TMM)*, 2022.

Abstract: Real-time and data-driven video stabilization using IMU sensors.

⇒ **"When LLMs step into the 3D World: A Survey and Meta-Analysis of 3D Tasks via Multi-modal Large Language Models".** Xianzheng Ma[5]*, Yash Bhalgat*, Brandon Smart*, **Shuai Chen**, Xinghui Li, Jian Ding, Jindong Gu, Dave Zhenyu Chen, Songyou Peng, Jiawang Bian, Philip Torr, Marc Pollefeys, Matthias Nießner, Ian D Reid, Angel X. Chang, Iro Laina, Victor Adrian Prisacariu. In *arXiv preprint: 2405.10255*, 2024

Abstract: A survey paper that provides a comprehensive overview of the methodologies enabling LLMs to process, understand, and generate 3D data.

⇒ **"Scene Coordinate Reconstruction: Posing of Image Collections via Incremental Learning of a Relocalizer".** Eric Brachmann, Jamie Wynn, **Shuai Chen**, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, Victor Adrian Prisacariu. In *European Conference on Computer Vision (ECCV Oral Presentation)*, 2024

Abstract: A novel learning-based structure-from-motion (SfM) method using scene coordinate regression.

[5]*Equal First Author Contribution.

Chapter 2

Literature Review

This chapter presents a review of the literature surrounding the problem of visual relocalisation (Section 2.1), with an exploration of its key approaches and advancements. Moreover, it provides overviews for novel view synthesis (Section 2.2), which is highly relevant to the contributions of this thesis. As this review is focused on the literature directly relevant to our work, we refer readers to more comprehensive surveys on visual relocalisation in [127, 157] and Section IV of [24], and on novel view synthesis in [156, 166, 39, 37].

2.1 Visual Relocalisation

Over the years many efforts in the literature have tackled the problem of visual relocalisation, and we have seen a rough demarcation of the types of approach into two main fields: the more traditional approaches, relying on geometric concepts and the estimation of correspondences between images and maps; and the more recent "direct" approaches, relying on neural networks to predict the absolute position and orientation of the image without an intermediate, explicit, matching step linking the 2D image realm with a 3D map of the scene. As illustrated in

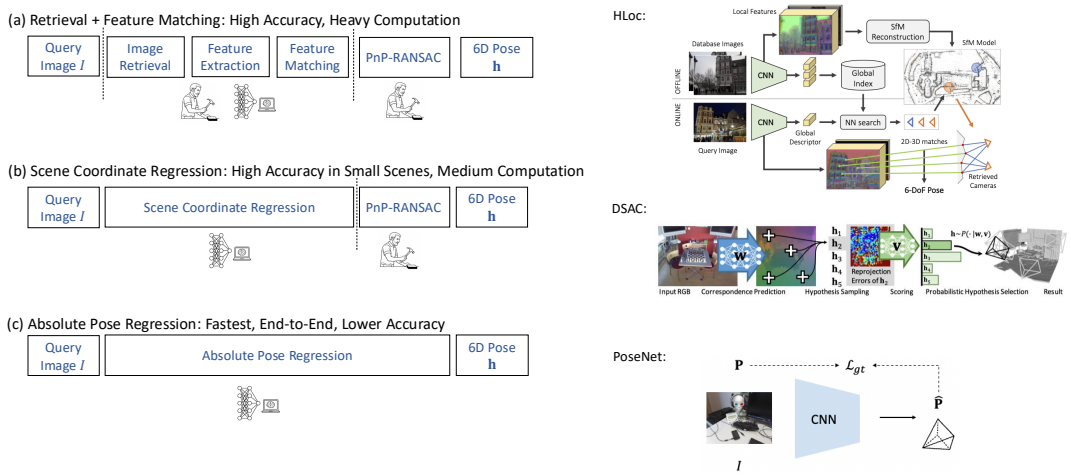


Figure 2.1: Overview and taxonomy of mainstream visual relocalisation paradigms. The figure illustrates three distinct technical routes for visual relocalisation: (a) Classical feature correspondences; (b) Scene coordinate regression; and (c) Absolute pose regression.

Fig. 2.1, these paradigms represent distinct trade-offs between accuracy, computational complexity, and inference speed. In the following sections (Section 2.1.1 and Section 2.1.2), we briefly explore the main approaches in each category.

We remind our readers not to confuse the task of visual relocalisation with other related visual localisation tasks, such as Structure from Motion (SfM) [122] and Simultaneous Localisation and Mapping (SLAM) [112]. While these tasks share common objectives and some methodologies for estimating camera poses from image data, visual relocalisation specifically aims to determine camera positions within a pre-defined global coordinate system, typically within an already mapped environment. In contrast, SfM usually operates offline on unordered image collections to reconstruct 3D structures and camera poses simultaneously, typically without real-time constraints. SLAM, on the other hand, incrementally estimates camera poses and simultaneously builds maps from scratch in real-time while navigating previously unknown environments.

2.1.1 Geometry-Based Visual Relocalisation

Geometry-based approaches rely on estimating correspondences between pixels in the query images and points in the scene's map. These correspondences effectively establish a 2D-to-3D matching that can be exploited by pose-solving methods such as PnP/RANSAC [57, 52] to compute the pose of the camera at the moment it captures the image.

Classical Feature Correspondences

There are several ways to estimate those correspondences. Classical approaches typically employ off-the-shelf detectors and descriptors, such as SIFT [105], SURF [6], or ORB [136], to match query image pixels directly to a pre-built 3D point cloud database generated by offline SfM [142, 143, 19, 71]. More recently, advanced neural-based methods have been developed, employing learned descriptors [135, 184, 41, 45, 134, 170], improved matchers [138, 164, 95, 26, 99, 3, 46, 47] and different map representations to estimate better correspondences from more challenging images or viewpoints [137, 107, 139, 123].

These approaches leverage the underlying geometric principles governing image formation and capture, yielding accurate pose estimations with low errors, often in the order of a few centimetres. One effective approach is to rely on the image retrieval method [1, 2, 140, 168, 178, 186, 62] to retrieve the most similar reference images from a pre-built database and establish the feature correspondences from the query images to the reference images to approximate camera poses [143, 165, 137, 139]. Alternatively, more direct methods [141, 73, 102, 59] skip explicit retrieval, directly associating query descriptors with descriptors of 3D points stored in the pre-built map.

However, these classical feature correspondences-based methods are not without drawbacks: they generally require the creation of an explicit map (*e.g.,* in the

form of a 3D point cloud created via SfM) of the scene ahead of time in order to associate the descriptors to 3D coordinates, and that is typically time-consuming and storage memory demanding when large-scale scene reconstruction is needed. Secondly, due to their explicit, multi-stage pipelines, they typically struggle to achieve real-time performance. The methods are also sensitive to the quality of other components in the pipeline, such as image retrieval, where inaccurate retrieval can significantly degrade pose estimation accuracy.

Scene Coordinate Regression

In recent years, a new approach to geometry-based relocalization became prominent: scene coordinate regression (SCR). In this scenario, the map of the scene is directly encoded in a fixed-size set of weights of a neural network. At localisation time, the query image is passed through the network, yielding per-pixel scene coordinates that can be directly used by a pose solver to estimate the camera pose [12, 13, 182, 14, 94, 43, 11]. While effective, these methods have typically required training a new network for every new target scene, potentially taking several hours [14], thus hindering their large-scale application.

Recently, an approach to scene coordinate regression that can take mere minutes to be trained for every scene was presented in [11], making practical deployment of SCR networks a possibility. As the correspondence-based methods mentioned above, coordinate regression approaches are also very accurate by relying on geometric information on the structure of the scene. Nevertheless, scene coordinate regression methods still require an explicit stage where a pose solver must process the correspondences generated by the method to estimate the camera pose. Conversely, Absolute Pose Regression methods, which constitute the primary focus of this thesis, do not have this requirement since the regressor network can go directly from image to pose in an end-to-end fashion.

In Chapter 8, we present an approach that belongs to the family of end-to-end APR methods. However, it also draws inspiration from SCR by using predicted scene coordinates as an intermediate representation. Unlike traditional SCR, instead of relying on a separate pose solver like in SCR, we employ a transformer-based regressor to estimate the pose directly, combining the strengths of both SCR and APR within a unified framework.

2.1.2 Absolute Pose Regression

Absolute Pose Regression approaches have also garnered notable attention, primarily due to their simplicity and efficiency. These methods directly predict camera poses via end-to-end neural networks and typically only require a few milliseconds to infer using a low-end consumer-level GPU (i.e., GTX 1080), thus being the current fastest visual relocalisation method.

Kendall et al. introduced the first APR model, PoseNet [79, 77, 78], where a feed-forward neural network directly regresses a 7-dimensional pose vector for every query image. Successive works explore diverse architectural designs such as hourglass networks [109], bifurcated translation and rotation regression [179, 117], attention layers [173, 152, 153, 154], and LSTM layers [172]. Like scene coordinate regression methods, APR approaches implicitly encode the map of the scene to a fixed set of weights regardless of the map size. One notable feature of the APR method, as illustrated in its pioneering work [79], is its robustness to low-quality queries such as motion blurs.

However, the performance of existing APR is still far behind other relocalisation approaches utilising geometric principles, typically of at least one order of magnitude in precision. We believe that the root cause is that the traditional APR formulation lacks an effective means to fuse 3D-based geometry-based priors and is prone to overfitting the training set [144]. Since the majority of the works in this

thesis focus on enhancing APR approaches, we present existing efforts to improve APRs in several aspects in the following sub-sections. For the sake of a thorough overview of this domain, we also include some of our publications presented in this thesis.

Loss Function and Optimisation

Balancing translation and rotation losses during network training is a critical challenge in APR. Kendall and Cipolla [78] addressed this by introducing learnable weights and a reprojection loss, incorporating some scene geometry into the training. Some works explore using different rotation representations, such as quaternion [79], 6D [194] or 9D [89] rotation representations.

Other research efforts attempt to improve APR performance with different optimisation strategies, such as relative pose constraints [15], uncertainty awareness [78, 115], multi-scene training [10, 152, 87], or a sequential formulation like temporal filtering [36, 115] and multitasking [131]. MapNet [15] trains the network using both absolute pose loss and relative pose loss, but can infer in a single-frame manner. Our work in Chapter 3, Direct-PoseNet, uses photometric losses inspired by direct matching [72, 161, 119, 50, 128, 49, 147] in visual odometry to refine pose estimates through RGB differences. Several follow-up works [30, 98, 192, 70], including DFNet (Chapter 4 of this thesis), have replaced photometric losses with more robust feature-metric supervision.

Semi-Supervised Learning

Semi-supervised methods such as MapNet [15] and Direct-PoseNet [31] aim to reduce dependency on labelled datasets. Rather than training only on images with ground-truth pose annotation, MapNet proposes to train on unlabelled video frames by adding pairwise geometric constraints between video frames using ad-

ditional VO algorithms [49, 50]. Direct-PoseNet (Chapter 3) also adopts a similar paradigm of leveraging additional unlabelled data, and finetunes the trained network using images from arbitrary viewpoints, by minimising the photometric loss between the rendered and unlabelled images. Both MapNet and Direct-PoseNet prove that these approaches enhance robustness and generalisation, and subsequent works like DFNet (Chapter 4) and PMNet [98] follow this school of thought.

Uncertainty Estimation

Recent APR studies incorporate uncertainty estimation to quantify the reliability of pose predictions and enhance robustness. Bayesian PoseNet [77] introduced Monte Carlo dropout, approximating uncertainty through multiple pose hypotheses, while AD-PoseNet [69] uses prior-guided dropout to derive a pose distribution. CoordiNet [115] complements these methods by learning heteroscedastic uncertainty and refining poses with an Extended Kalman Filter. Deng *et al.* [40, 18] and Zangeneh *et al.* [187] use pose distributions to represent uncertainty. These uncertainty-aware (UA) APR approaches have been proven to be effective during both training and inference, particularly in smoothing trajectories for long-term tracking. However, they often increase computational overhead [77, 69], require hyperparameter tuning [18], and fail to match the pose accuracy of most recent non-uncertainty-aware (non-UA) methods. In contrast, our work in Chapter 6 proposes a lightweight uncertainty estimation module aiming to assist the state-of-the-art APR pose refinement method with minimum computational overhead (under 6 milliseconds) and without altering the original APR architecture or training schemes.

Novel View Synthesis-based Data Augmentation

Novel View Synthesis (NVS) can be beneficial to the visual relocalisation task. For example, NVS can expand the training space by generating synthetic data. Purkait *et al.* [130] propose a method to generate realistic synthetic training data for pose regression by leveraging the 3D map and the feature correspondences. However, this method relies on a precomputed reconstructed 3D map, which is not universally available and may often require a long time to reconstruct. LENS [114] deploys a NeRF-W-inspired model [108], which can be viewed as an implicit replacement for the explicit 3D map, to sample the scene boundaries and synthesise virtual views with uniformly generated virtual camera poses. LENS is limited by its costly off-line computation efficiency and the inherent domain gap between synthetic and real images, *e.g.,* caused by changes in illumination, dynamic objects, or artefacts. Building on this idea, DFNet (Chapter 4) uses a customised NeRF to generate realistic exposure changes in view synthesis, beginning to address the domain adaptation problem between synthetic and real images. It is important to note that using NVS as an effective data augmentation tool has an impact beyond APR approaches, for example, in SCR [27, 17] and visual-inertial odometry (VIO) [65].

2.2 Novel View Synthesis

Novel View Synthesis aims to generate new views of a scene from a limited set of input views. Early methods treated this task as image-based rendering (IBR), relying on interpolation or extrapolation techniques to fill in missing viewpoints from existing images [28]. Such techniques often required dense sampling of the scene [63, 90, 16] or the creation of explicit proxy geometries [149]. The advent of multi-view stereo [150, 83, 61, 55, 146] and structure-from-motion-based ap-

proaches [104, 167, 66, 160, 145] enabled more robust 3D reconstructions, providing a stronger geometric basis for NVS [22, 197, 48, 67, 82].

However, classical NVS pipelines frequently suffer from artefacts due to erroneous depth maps or incomplete geometry, especially in regions with insufficient feature correspondence or occlusions [38]. To tackle these issues, learning-based techniques and implicit 3D representations [33, 110, 124, 159] were later introduced, serving as critical stepping stones for more recent NVS methods. Building on these advancements, methods such as Neural Radiance Fields and 3D Gaussian Splatting have emerged and led significant progress in this field.

Since NeRF and 3D-GS are the most relevant NVS approaches to the works presented in this thesis, we will focus our review on these two methods in the following sections. For a more comprehensive review of other NVS techniques, we refer to the surveys cited at the beginning of Chapter 2.

Neural Radiance Fields and Neural Feature Fields

Recent research in Neural Radiance Fields (NeRF) [111, 108, 190, 177, 96, 129, 148, 120, 132, 125, 180, 9, 8, 4, 5] led to a major paradigm shift for NVS by enabling the learning of continuous, implicit 3D volumetric representations directly from sparse sets of input images. By employing multi-layer perceptrons (MLPs) to encode colour and density as a function of 3D coordinates and viewing direction, NeRF can produce highly photorealistic renderings through differentiable volume rendering techniques.

Building upon NeRF, subsequent studies have extended the model beyond the RGB space to operate directly in feature space. Recent extensions of NeRF predict and render *feature fields* in addition to the traditional density and appearance fields. These feature fields are typically learnt using supervision from a 2D feature extractor and are integrated into the volumetric rendering framework. Studies such

as [169, 81, 7] demonstrated that these 3D feature fields outperform conventional 2D baselines [21, 91, 34] in downstream tasks such as 2D object retrieval and 3D segmentation. Additionally, CLIPFields [151] has introduced feature fields as scene memory for applications like robot navigation.

Our work in Chapter 3 and Chapter 4 is among the first to design NeRF-based models specifically tailored for training APRs. In Chapter 5 and Chapter 6, we further extend this line of research by applying a customised NeRF to support test-time pose refinement.

Inverting NeRF

Inverting NeRF for camera pose estimation has emerged as a popular research area, aiming to leverage NeRF's capabilities for determining camera positions and orientations. The intuition is that each camera pose in NeRF corresponds to a unique scene rendering, thus enabling the inverse mapping for pose estimation. Yen-Chen et al. [183] introduced iNeRF, a framework that inverts a pre-trained NeRF model for pose estimation. Starting with an initial pose, iNeRF iteratively refines the pose by minimising the residual between rendered pixels from the NeRF model and observed image pixels through gradient descent. However, this approach requires iterative optimisation for each test image and is constrained to specific scenarios. The concurrent work Direct-PoseNet [31] demonstrated that an inverted NeRF can be applied to assist in the training of the APR network. NeRF$--$ and BARF [177, 97] advanced the concept by proposing joint optimisation strategies, where camera poses are treated as learnable parameters during the training of NeRFs in non-360° scenes. These methods provide a theoretical framework for effectively utilising NeRFs for pose optimisation.

Follow-up works, such as NICE-SLAM and iMAP [196, 162], utilise NeRF for dense geometry and real-time camera tracking. DFNet and PMNet [30, 98] extend

Direct-PoseNet with robust feature metric matching to adapt to the real exposure change in view synthesis.

Despite these advancements, challenges remain to improve convergence speed and accuracy. For example, NeRF-based pose estimation methods often rely on time-consuming iterative rendering and pose updates, leading to slow convergence and limited accuracy. Although our work in Chapter 5 further improves 59%+ accuracies comparing to previous APR SOTA by applying neural feature fields to the post-processing stage, NeRF-based pose estimation methods still suffers from long refinement runtime, and face difficulties in surpassing results compare to the best of SCR-based methods. In Chapter 6, we propose a new hierarchical pose refinement framework, HR-APR, to speed up the camera pose refinement speed by up to 27.4%, but the average runtime of each query is still in the range of several seconds on a high-performance GPU. Other NeRF-based methods like FQN [60], CrossFire [113], NeRFLoc [100], and NeRFMatch [193] improve pose estimation accuracy by establishing 2D-3D matches but require specialised feature extractors and suffer from slow rendering and quality issues.

3D Gaussian Splatting and Inverting 3D-GS

Although accelerating NeRFs [101, 118, 58, 133, 185, 23, 116, 53, 163, 32] has become an active research area in the 3D vision community, many of these approaches require trade-offs between rendering quality and computational efficiency. In contrast, 3D Gaussian Splatting [80, 106, 86] has emerged as a competitive alternative, offering both high-quality rendering and efficient speed.

The efficiency of 3D-GS comes from its unique representation. 3D-GS extends traditional point cloud representations by associating each point with additional attributes that model the radiance emitted in the surrounding spatial region using anisotropic 3D Gaussian "blobs". These 3D Gaussians are typically initialised

from SfM point clouds [145], and optimised using differentiable rendering. 3D Gaussian Splatting achieves the state-of-the-art NVS result at a fraction of NeRF computation using efficient rasterisation [84].

Similarly to inverting NeRF methods, inverting 3D-GS also demonstrates a positive impact on pose estimation. By replacing NeRF with 3D-GS, pose estimation methods can render highly detailed images with minimal computation. This dramatically improves the usability of renderer-based visual localisation. Several methods [54, 75, 93] explore efficient 3D-GS reconstruction with unposed images. [85] integrates 3D-GS into the Structure-from-Motion framework. [76, 68, 126, 188, 64] develop 3D-GS-based SLAM system.

Our work in Chapter 7, along with several other concurrent and follow-up studies [74, 121, 158, 35, 195, 189, 70], is among the first to employ 3D-GS in relocalisation pipelines. Compared to concurrent work, our proposed method consistently achieves SOTA accuracies across multiple benchmarks and works on top of APR and SCR-based methods.

Part I

Training of Absolute Pose Regression

Chapter 3

Direct Photometric Matching with NeRF

This chapter begins to address the limitations of Absolute Pose Regression-based visual relocalisation methods, which typically suffer from limited accuracy, poor generalisation, and a heavy reliance on large volumes of labelled image-pose pairs. These constraints hinder their applicability in real-world applications. For example, generating such labelled data often requires computationally intensive methods like SfM. Meanwhile, unlabelled images, which are frequently available in practical scenarios, remain underutilised.

To overcome these limitations, we introduce Direct-PoseNet, a novel training pipeline that integrates 3D geometric reasoning via photometric supervision. Specifically, the method incorporates a NeRF-based differentiable rendering module, enforcing photometric consistency between observed and rendered views of a scene during training. This supervision complements traditional pose regression loss, enabling the network to implicitly encode geometric constraints and improve localisation accuracy.

Upon its publication at the *2021 International Conference on 3D Vision*,

Direct-PoseNet achieved state-of-the-art APR performance on benchmark datasets, including 7-Scenes and LLFF. Ablation studies demonstrate that photometric supervision and semi-supervised learning are the key factors driving these improvements. To the best of our knowledge, Direct-PoseNet is the first method to demonstrate that a differentiable renderer can enhance pose regression performance.

Direct-PoseNet: Absolute Pose Regression with Photometric Consistency

Shuai Chen Zirui Wang Victor Prisacariu
Active Vision Lab, University of Oxford
{shuaic, ryan, victor}@robots.ox.ac.uk

Abstract

We present a relocalization pipeline, which combines an absolute pose regression (APR) network with a novel view synthesis based direct matching module, offering superior accuracy while maintaining low inference time. Our contribution is twofold: i) we design a direct matching module that supplies a photometric supervision signal to refine the pose regression network via differentiable rendering; ii) we show that our method can easily cope with additional unlabeled data without the need for external supervision such as traditional visual odometry or pose graph optimization. As a result, our method achieves state-of-the-art performance among all other single-image APR methods on the 7-Scenes benchmark and the LLFF dataset.

1. Introduction

Camera localization is a classical problem in computer vision and robotics. It is a core component for many applications such as virtual and augmented reality, indoor navigation systems, and autonomous driving. A typical visual-based localization algorithm is designed to determine the camera's 6-DOF positions and orientations from taking as input an RGB or RGB-D image.

The classical approach to solve this problem is built upon finding 2D-3D correspondences [2, 3, 5, 47, 48, 51, 55] between 2D image position and 3D points in space. Then an n-point pose (PnP) solver is applied to the 2D-3D matches inside a RANSAC [4, 8, 12, 44] loop. Traditionally, 2D-3D matches can be found using local feature descriptor matching, and many approaches require depth or structure-from-motion (SfM) reconstruction to build robust 3D geometric correlations [46, 48]. Recent methods use machine learning to regress 3D scene coordinates from image patches [2, 3, 5, 51] directly. Overall, 3D structure-based methods still achieve state-of-the-art (SOTA) accuracy, as discussed by Sattler *et al*. [49]. However, the presence of highly accurate depth images or SfM models is not universally available in real-life applications, especially for many consumer-grade devices such as smartphones or tablets. Most deep 3D structure-based methods are computation resources intensive and cannot easily achieve real-time inference with SOTA accuracy constraints.

Another line of approaches is deep learning-based pose regression [6, 20, 21, 22, 30, 35, 43, 59, 58, 61], also known as absolute pose regression (APR). These approaches propose to train a scene-specific deep neural network to predict 6-DoF camera pose relative to a scene directly from images. Despite obtaining inferior performances in localization benchmarks, it has gained popularity due to its high efficiency and simplicity by learning the full localization pipeline in a Convolutional Neural Network (CNN). The end-to-end approach has several appealing features compared to 3D structure-based methods: (1) most APR algorithms display great portability for commercial deployment at applications where fast and reliable performance is crucial. For example, the groundwork PoseNet [22] runs its entire process in less than $6ms$. (2) the CNN only requires RGB images input and does not rely on depth maps or SfM reconstructions, which is less hardware constrained. (3) it keeps a low memory footprint in megabytes regardless of scene sizes.

Despite these benefits, drawbacks of the APR method are also apparent. It is known to be prone to overfit the training set and significantly less accurate than structure-based methods, as shown in Sattler *et al*. [49] and Shavit *et al*. [52] . Both studies suggest that scene geometry is key for obtaining accurate pose estimation. Prior efforts have tried to add geometric constraints by finding relative pose [1, 6, 43, 58] or using reprojection error [21]. Nevertheless, it is clear that existing single image APR solutions are not yet able to compete with structure-based methods.

We address the problem of single-image APR by introducing direct matching supervision inspired by direct Visual Odometry (VO) approaches [10, 11]. The key intuition is that the predicted pose error is inversely proportional to the visual similarity between the query image and a rendering of the 3D scene of the relocalized pose. The proposed APR framework improves pose regression using direct matching supervised by the photometric similarity between the input query image and the rendered image of the scene using the

predicted pose. In the testing stage, our method runs like a standard APR method without extra computational cost. To our knowledge, this paper is the first camera pose regression method to use direct matching/photometric supervision. We summarize our contributions as follows:

- We introduce a novel camera relocalization pipeline consisting of a pose regression network and a direct matching module such that network learning is supervised by not only the traditional pose regression loss, but also a photometric loss.
- We show how unlabeled images can be leveraged using photometric loss in the direct matching module to further improve the pose regression performance without extra supervision, such as relative pose constraints.

With contributions above, our method achieves state-of-the-art performance in single-image APR on 7-Scenes benchmark and LLFF dataset.

This paper is organized as follows: we introduce existing APR methods and other related work in Section 2. Our relocalization pipeline is detailed in Section 3, with experimental results and analysis discussed in Section 4. Section 5 concludes our work.

2. Related Work

Absolute Pose Regression Absolute pose regression methods typically require a CNN classifier that has been pre-trained from the image classification dataset. It then uses transfer learning to fine-tune the feature extractors to regress the camera pose from one or more given image sequences. To get a thorough review in this area, the interested reads is refered to Sattler *et al*. [49].

The common practice in this area is introduced by PoseNet [22]. A simple pose regressor can take an arbitrary RGB image as the input and learn to regress the correspondent camera position and orientation. Successors of PoseNet focus to improve the framework in several aspects. [30, 59, 61] seek to enhance network architectures. LSTM PoseNet [59] combines LSTM with CNN to reduce feature dimensions for pose regression. Hourglass PoseNet [30] adapts an encoder-decoder style backbone. BranchNet [61] uses a multi-task CNN where low-level common features are extracted before splitting the network into two separate branches to predict the camera position and orientation. Kendall and Cipolla [21] proposed learnable weights to sidestep hyperparameter tunning that balances the translation and rotation loss in PoseNet. Besides learning the optimal weight between losses, they attempt to leverage scene geometry for pose regression by formulating a reprojection error between the ground-truth pose and predicted pose. While we share the same insight that geometry would help pose regression, we differ from theirs in that 1) the scene geometry is implicitly represented in a novel view synthesis-enabled direct matching module; 2) we introduce a dense pixel-level photometric supervision.

Unlabeled training in APR Rather than only training on images with ground-truth pose annotation, MapNet [6] is able to train on unlabeled video frames by adding pairwise geometric constraints between video frames using additional VO algorithms [10, 11]. During inference, it can utilize pose graph optimization (PGO) for post-processing to further boost the performance. We also recognize the importance of using additional unlabeled data in this work. However, instead of acquiring constraints from an additional VO algorithm or PGO, which assumes video frames are available, our method can train on images from arbitrary viewpoints by minimizing the photometric loss between the rendered images and the input images.

Direct Matching for Motion Estimation Direct matching methods or direct methods refer to the commonly used methods to recover camera motions by directly measuring image intensities [18] in VO and simultaneous localization and mapping (SLAM) systems. In contrast to the feature-based methods [9, 19, 24, 25] that minimize reprojection errors based on corresponding features among frames, direct methods exploit all information in the image and recover camera motions by minimizing the photometric error. Compared to feature-based methods, direct methods are often more reliable in sparse textured environments and do not have feature extraction operations to add in computation costs. DTAM [36], REMODE [42], and LSD-SLAM [11] employ dense reconstruction using direct methods. SVO [13, 14] proposes a hybrid approach to implement direct VO motion estimation at the SLAM frontend and uses feature-based methods in the backend mapping thread. DSO [10] further proposes a sparse and direct method. This paper is partly inspired by direct methods and adopts photometric supervision in training the pose regression network. However, our method is different from direct VO methods in several aspects. First, our method does not compute photometric errors based on image intensities but RGB differences. Second, our method provides absolute pose estimation using a single image, but the methods above are designed to take a pair of neighboring frames for computing relative motion.

Novel View Synthesis Novel view synthesis (NVS) is a long-standing problem in computer graphics. It aims to generate novel camera perspectives based on image samples of the scenes [56]. Early works in this field can be traced back to nearly 30 years ago when some required using densely captured views of the scene [17, 26], and others [7] interpolate novel views using image warping. Recently, novel view

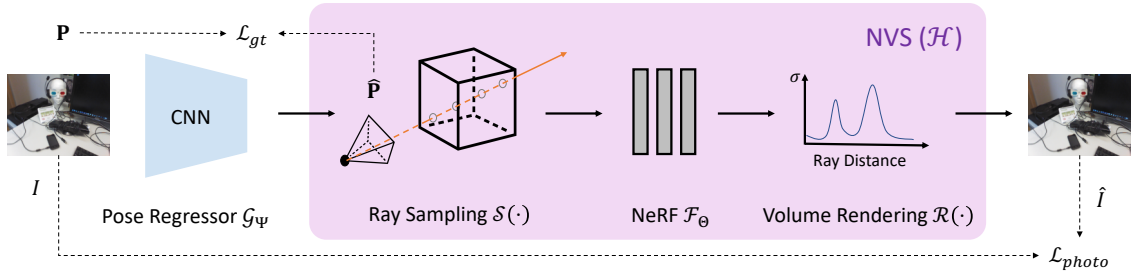


Figure 1: Overview of our proposed training pipeline. Given an input image I, the pose regressor $\mathcal{G}_\Psi$ predicts a pose estimation $\hat{\mathbf{P}}$, from which the NVS system $\mathcal{H}$ renders a synthetic image $\hat{I}$, supplying our direct matching supervision signal $\mathcal{L}_{photo}$ and refining $\mathcal{G}_\Psi$ along with the ground truth supervision $\mathcal{L}_{gt}$.

synthesis has made rapid progress in achieving photorealistic view synthesis [27, 28, 29, 32, 34, 39, 41, 50, 53, 57] with sparser view samples, thanks to recent development in neural 3D shape representation [31, 37, 38, 54]. We select a recent popular approach from Mildenhall *et al*. [34] to build our camera re-localization training pipeline in this work. Specifically, we incorporate NeRF architecture to provide photometric supervision to the pose regression model. We consider two other NeRF-based works that are able to estimate camera pose related to our paper: iNeRF [63] estimates pose iteratively by inverting a pre-trained NeRF model on the test images. Wang *et al*. [60] show that 3D scene representation and camera poses can be jointly optimized within a NeRF framework. Nonetheless, the fundamental difference of the proposed method to theirs is that we only use differentiable rendering to compute photometric loss in training of our pose regressor. After training, we are able to predict camera pose in a single network forward pass, whereas their methods require iterative optimization in test time.

3. Method

Fig. 1 illustrates our proposed relocalization pipeline, which consists of an NVS-enabled direct matching module and a pose regression network. We aim to predict a camera pose $\hat{v}$ for an input image I via a pose regression network $\mathcal{G}_\Psi$, which is supervised by a novel direct matching signal along with ground truth poses during training. At test time, only the pose regression network is required, ensuring rapid inference while offering superior relocalization accuracy.

This section is organized as follows: Section 3.1 details our direct matching module. A full system setup is explained in Section 3.2. To explore the possibility of utilizing more data, a further unlabeled training scheme is explained in Section 3.3.

3.1. Direct Matching

Direct matching is a common approach in traditional SLAM and VO systems [10, 11, 13, 14, 36, 42]. It refers to the process that optimizes camera poses via minimizing a photometric loss. In this work, we adapt the direct matching concept and apply it to assist the training of our pose regression network. Concretely, assuming a pre-trained pose regression network $\mathcal{G}_\Psi$ and an NVS system $\mathcal{H}$ are available, given an image I captured at viewpoint v and its pose estimation $\hat{v} = \mathcal{G}_\Psi(I)$, our direct matching module constrains Ψ via minimizing the photometric difference $\mathcal{L}_{photo}(\hat{I}, I)$ between a synthetic image $\hat{I} = \mathcal{H}(\hat{v})$ rendered by the NVS model $\mathcal{H}$ at viewpoint $\hat{v}$, and its true observation I:

$$\mathcal{L}_{photo}(\hat{I}, I) = \|\hat{I} - I\|_2. \tag{1}$$

Any NVS system with a differentiable renderer could be selected as the NVS system $\mathcal{H}$. For simplicity, we choose the NeRF [34] in this work, due to the high quality reconstructions it produces.

From a high-level point of view, NeRF-based NVS requires three main components: 1) a neural radiance field function $\mathcal{F}_\Theta$ that models a 3D volume, 2) a differentiable volume renderer $\mathcal{R}(\cdot)$ that enables back-propagation, and 3) a viewpoint-dependent volume sampling function $\mathcal{S}(\cdot)$ that provides 3D sample locations and view directions for $\mathcal{F}_\Theta$ and $\mathcal{R}(\cdot)$ given a camera pose. Consequently, an image rendered from an NVS system $\mathcal{H}$ using NeRF can be formulated by:

$$\hat{I}_v = \mathcal{H}(v) \triangleq \mathcal{R}(\mathcal{F}_\Theta, \mathcal{S}(v)), \tag{2}$$

where $\hat{I}_v$ denotes a synthetic image rendered at a pose v, and all operations above are differentiable. As a result, our direct matching system updates Ψ by minimizing photometric loss in Eq. (1).

3.2. System Setup

Training Pipeline As mentioned above, our system consists of two main components, a pose regression network

$\mathcal{G}_\Psi$ and an NVS-enabled direct matching module. With each component pre-trained on target data, we join them together to further refine the pose regression network: given an input image I, the pose regression network predicts a pose $\hat{v}$, from which an NVS system $\mathcal{H}$ renders a synthetic image $\hat{I}_{\hat{v}} = \mathcal{H}(\hat{v})$, enabling our direct matching supervision $\mathcal{L}_{photo}$. Meanwhile, the ground truth supervision $\mathcal{L}_{gt}$ is applied as well. As a result, the pose regression network $\mathcal{G}_\Psi$ is refined through a back-propagation of a weighted sum of $\mathcal{L}_{photo}$ and $\mathcal{L}_{gt}$ (Eq. (3)). Mathematically, our training pipeline can be framed as:

$$\Psi^* = \arg\min_{\Psi}(\lambda_1 \mathcal{L}_{photo} + \lambda_2 \mathcal{L}_{gt}), \tag{3}$$

where Ψ^* denotes optimized network parameters, and λ_1, λ_2 denote weights for each loss terms, respectively.

Pose Regression Network Our pose regression network $\mathcal{G}_\Psi$ follows the line of PoseNet [22] work, including a pre-trained feature extractor backbone and a fully connected layer that outputs a camera pose matrix $\hat{\mathbf{P}}$. Prior works [6, 20, 21, 22, 30, 43, 58, 59] typically use a quaternion or an axis-angle representation during pose estimation, requiring balancing between rotation and translation terms. Instead, our network regress a camera pose v using the representation of $\mathbf{P} = [\mathbf{R}|\mathbf{t}]$ to overcome this issue, where $\mathbf{R} \in \mathrm{SO}(3)$ is a rotation matrix denotes a camera orientation and $\mathbf{t} \in \mathbb{R}^3$ denotes camera position. For clarity, we refer v as a general pose concept and $\mathbf{P}$ as a specific pose representation.

Our ground truth supervision loss is defined as the L2 distance between a ground truth pose $\mathbf{P}$ and an estimated pose $\hat{\mathbf{P}}$:

$$\mathcal{L}_{gt} = \|\mathbf{P} - \hat{\mathbf{P}}\|_2, \tag{4}$$

removing the need for balancing rotation and translation terms while offering competitive performance.

NeRF Two challenges arise while adapting NeRF as our NVS system $\mathcal{H}$ in relocalization context. First, the training cost is expensive in relocalization tasks, where a training video could easily yield thousands of images even after frame subsampling. We resolve this issue by reducing the NeRF model size and removing the hierarchical training scheme.

Second, strong artifacts occur in synthetic images if the NeRF is applied in our task without modifications. Two reasons account for that: a) NeRF is not designed for outward-looking scenes [64] and b) photometric consistency is violated when auto-focus/exposure fluctuation and rolling shutter effect appear. We mitigate this issue by adapting a coarse-to-fine positional encoding scheme $\gamma_\alpha(\cdot)$ from Nerfies [39]. Specifically, an input signal p is encoded by $\gamma_\alpha(p) = [p, \ldots, w_k(\alpha_t)\sin(2^k\pi p), w_k(\alpha_t)\cos(2^k\pi p), \ldots]$, where $0 \leq k \leq m-1, m \in \mathbb{N}$ and $w_k(\alpha_t)$ activates each band over epoch t, controlled by $\alpha_t = mt/N$. N is a user-defined maximum epoch number in training, where k reaches the maximum frequency band $m-1$. We refer interested readers to the full mathematical expression for $w_k(\alpha_t)$ in our supplementary material.

With the progressive positional encoding function, we are able to adapt NeRF as our NVS system $\mathcal{H}$, reducing artifacts and preserving high frequency details. Please refer to Section 4.4 for more discussion on the effectiveness in this approach.

3.3. Unlabeled Training

Inspired by MapNet+ [6], we propose to improve pose estimation in a semi-supervised manner, with unlabeled sequences captured in the same training scene. Unlike [6], which enforces a relative geometric constraint between two nearby frames and requires an additional VO algorithm, we rely on our bootstrapped pipeline to further refine the pose regression network $\mathcal{G}_\Psi$. Given an input image without ground truth pose annotation but not too far from labeled training videos, the training of $\mathcal{G}_\Psi$ can be supervised by the photometric loss between the synthetic image rendered by the direct matching module using the predicted pose. This semi-supervised training scheme can be effectively set up by setting $\lambda_1 = 1.0$ and $\lambda_2 = 0.0$. We find our unlabeled training works well, evidenced by the performance in Table 2.

4. Experiments

In the following, we discuss the implementation details of our solution in Section 4.1. We perform a thorough evaluation of the proposed method in Section 4.2 on the 7-Scenes dataset. We further evaluate our method on the LLFF dataset to demonstrate that the proposed method benefits from both the traditional pose regression method and the direct matching method (Section 4.3). Finally, to gain more insights to our modification on positional encoding and the effectiveness of direct matching for camera localization, we apply more experiments in the ablation study (Section 4.4).

4.1. Implementation Details

Pose Regression We build our pose regression model upon prior DNN-based methods [6, 21, 22] using PyTorch [40]. We choose to use the MobileNetV2 [45] backbone in this work. We freeze the batch normalization layers [16] from the pre-trained ImageNet backbone to train our baseline pose regression model. Since a direct rotation matrix regression may not belong to SO(3), a singular value decomposition (SVD) is applied to normalize the rotation component of $\hat{\mathbf{P}}$ during inference time. However, we also

	without unlabeled data									with unlabeled data		
Scene	PN [22]	PN learned weights [21]	geo. PN [21]	LSTM PN [59]	Hourglass PN [30]	BranchNet [61]	DSO [10]	MapNet [6]	Direct-PN	MapNet+ [6]	MapNet+ PGO [6]	Direct-PN+U
Chess	0.32/8.12	0.14/4.50	0.13/4.48	0.24/5.77	0.15/6.17	0.18/5.17	0.17/8.13	0.08/3.25	0.10/3.52	0.10/3.17	0.09/3.24	0.09/2.77
Fire	0.47/14.4	0.27/11.8	0.27/11.3	0.34/11.9	0.27/10.8	0.34/8.99	0.19/65.0	0.27/11.7	0.27/8.66	0.20/9.04	0.20/9.29	0.16/4.87
Heads	0.29/12.0	0.18/12.1	0.17/13.0	0.21/13.7	0.19/11.6	0.20/14.2	0.61/68.2	0.18/13.3	0.17/13.1	0.13/11.1	0.12/8.45	0.10/6.64
Office	0.48/7.68	0.20/5.77	0.19/5.55	0.30/8.08	0.21/8.48	0.30/7.05	1.51/16.8	0.17/5.15	0.16/5.96	0.18/5.38	0.19/5.42	0.17/5.04
Pumpkin	0.47/8.42	0.25/4.82	0.26/4.75	0.33/7.00	0.25/7.0	0.27/5.10	0.61/15.8	0.22/4.02	0.19/3.85	0.19/3.92	0.19/3.96	0.19/3.59
Kitchen	0.59/8.64	0.24/5.52	0.23/5.35	0.37/8.83	0.27/10.2	0.33/7.40	0.23/10.9	0.23/4.93	0.22/5.13	0.20/5.01	0.20/4.94	0.19/4.79
Stairs	0.47/13.8	0.37/10.6	0.35/12.4	0.40/13.7	0.29/12.5	0.38/10.3	0.26/21.3	0.30/12.1	0.32/10.61	0.30/13.4	0.27/10.6	0.24/8.52
Average	0.44/10.44	0.24/7.87	0.23/8.12	0.31/9.85	0.23/9.53	0.29/8.30	0.26/29.4	0.21/7.77	0.20/7.26	0.19/7.29	0.18/6.55	0.16/5.17

Table 1: Pose regression results on 7 Scenes datasets. We compare our method with both direct matching and absolute pose regression methods, in median translation error (m) and rotation error (°). Bottom row is the average of median errors of all scenes. PN denotes PoseNet. Numbers in red represent the best performance with or without unlabeled data.

find that the pose regression network learns to predict orthogonal rotation matrices even without using the SVD. All models are optimized with the Adam optimizer [23]. The base model is trained with a batch size of 4 and a learning rate of 1×10^{-4}. We implement an early stopping strategy with a patience value of 200 and schedule the learning rate decay for every 50 epochs on validation loss plateau with a factor of 0.95.

NeRF Our NeRF model is trained with input poses in SE(3). To ensure a consistent coordinate system in pose regression and NVS, we further align and recenter the camera poses with zero-means similar in Mildenhall *et al*. [34]. The NeRF architecture mainly follows the original implementation [34], except we apply a coarse-to-fine positional encoding $\gamma_\alpha(p)$ for both positions and directions. We set the maximum frequency band $m = 8$, and time to reach the maximum frequency band $N = 1200$ epochs.

Training For our proposed methods in Section 3.2 and Section 3.3, we train the pose regression model with direct matching (Direct-PoseNet) with $\lambda_1 = 0.3$ and $\lambda_2 = 0.7$, and further fine-tune the model (Direct-PoseNet+U) with $\lambda_1 = 1.0$ and $\lambda_2 = 0.0$ to simulate the unlabeled data circumstances. We set the batch size to 1 for training both models. The learning rate is set to 1×10^{-5} with the same early stopping strategy as above. All models are trained within 24 hours with a single Nvidia 1080Ti graphic card. In our experience, the NeRF training time can reach approximately the same as training PoseNet models. More details on network architecture and training procedure are provided in the supplementary material.

4.2. Evaluation on 7-Scenes

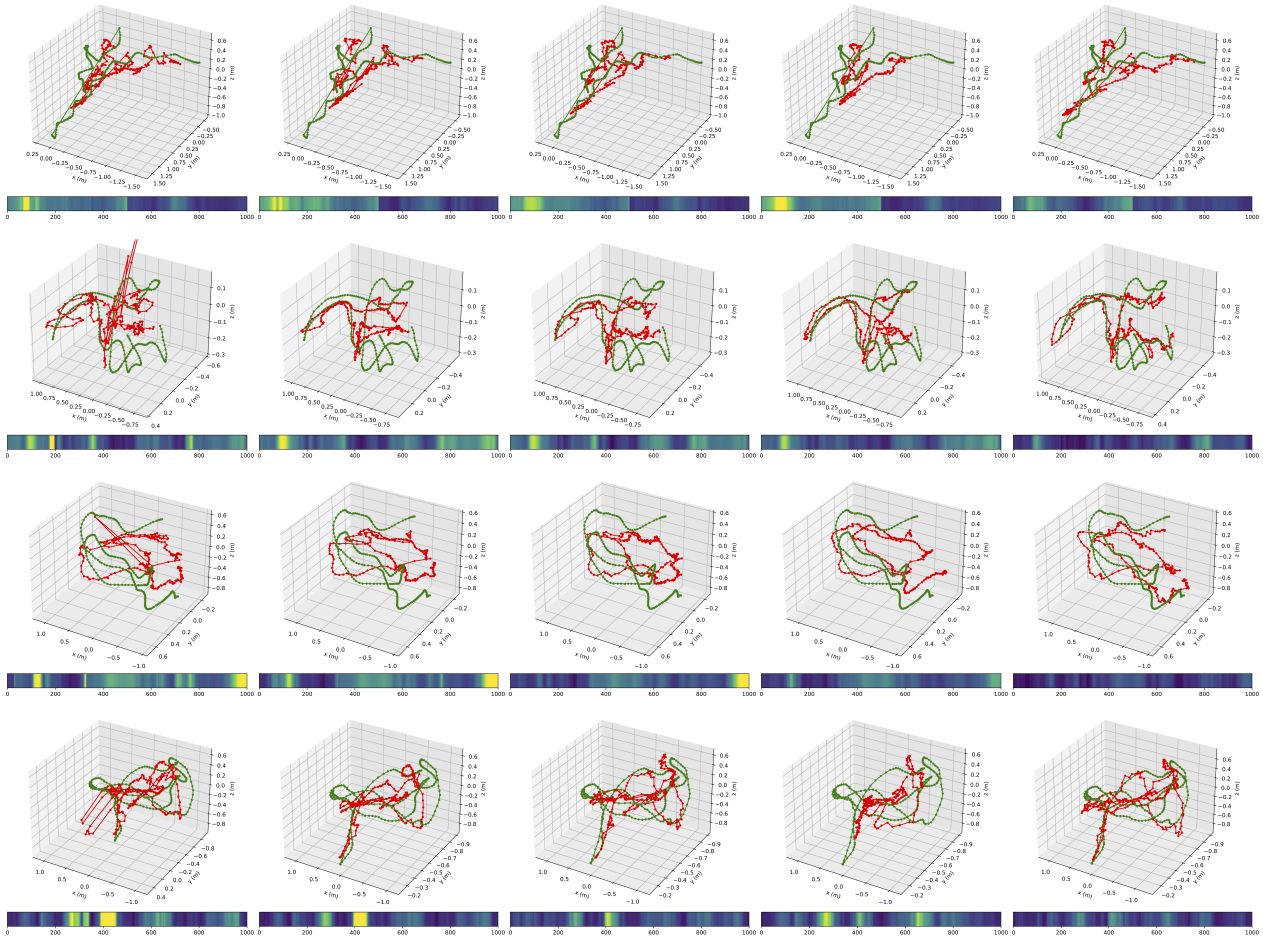
We evaluate our method on a well-known camera localization dataset 7-Scenes [15, 51]. It consists of seven indoor scenarios, each scale from $2m^3$ to $6m^3$. The sequences were shot by Kinect RGB-D camera at 640×480 resolution, and the ground truth poses were obtained by a dense 3D model.

The pose regression network takes an input image in 320×240 and the pre-trained NeRF model is trained with resized images in 160×120, but inference in 320×240 for Direct-PoseNet and Direct-PoseNet+U training. For each scene, we train our NeRF model with a learning rate of 5×10^{-4} for 4000 epochs with Adam optimizer and decays exponentially to 8×10^{-5} throughout the course of optimization. We set the near and far bounds $[b_n, b_f]$ to $[0.5, 4]$ except for the Heads scene, we set them to $[0.5, 2.5]$. However, unlike the original NeRF [34], which uses a coarse-to-fine sampling approach in its architecture, we only use a single MLP model with a width of 128. We sample one image with a batch of 1024 rays for each iteration, and each ray uniformly samples $N = 128$ bins. The above modifications achieve approximately $3\times$ speed up compare to the original NeRF paper. To further improve training efficiency, our NeRF model, Direct-PoseNet model, and Direct-PoseNet+U model only use a spacing window $d = 5$ of the training set for scenes contain ≤ 2000 frames, and $d = 10$ of the training set otherwise.

We summarize complete quantitative comparisons of our proposed method with prior absolute pose regression works and DSO in Table 9. For the experiment of Direct-PoseNet+U, we follow MapNet+ to use the unlabeled test sequences for fine-tuning. We do not use the entire test sequences for fine-tuning, but only 1/5 or 1/10 of the sequences described above to ensure our method is not overfitting to the entire test sequences. We also demonstrate a selection of the visual comparisons in Fig. 2.

4.3. Evaluation on LLFF

We further evaluate our method on another real-world complex scene dataset, the LLFF dataset [33]. The dataset consisting of 8 forward-facing scenes captured with a handheld cellphone and holds out $1/8$ of the data as the test set.



(a) PoseNet [6, 21, 22] (b) MapNet [6] (c) Direct-PoseNet (d) MapNet+PGO [6] (e) Direct-PoseNet+U

Figure 2: Visualization of camera relocalization results on 7-Scenes dataset [15, 51]. For each 3D plot, we show the ground truth camera trajectory in green and the predicted trajectory in red. The bottom color bar represents rotation errors for each subplot. Yellow represents high rotation error, and blue represents low rotation error for each test sequence. Sequence names from top to bottom are: Stairs-all, Heads-all, Fire-seq-03, Office-seq-09.

It is ideal for an experiment because a high-quality NeRF can be trained on LLFF to examine the combined effects of pose regression and direct matching supervision.

We compare our method with both the pose regression method and an inverting NVS method iNeRF by Lin *et al*. [63]. The iNeRF uses an iterative optimization approach on each test image to recover the camera pose by inverting a trained NeRF model. On the other hand, our method does not rely on iterative optimization and produces a more generalized and efficient model. To ensure a fair comparison, we trained our NeRF model to follow the same setting with Lin *et al*., which uses a standard NeRF model with a ray batch size of 2048. We fine-tune the NeRF model on four scenes (*Fern*, *Fortress*, *Horn*, *Room*) with the baseline pose regression model and compute the percentage of predicted pose whose error is less than 5cm and the percentage of predicted pose whose error is less than $5°$. We report the experiment results in Table 2. We observe that our proposed pose regression method gains benefit both from the

error rate in %	iNeRF	PoseNet+SE(3)	Direct-PoseNet+U
<5cm, <5°	73%, 71%	57%, 100%	78%, 100%

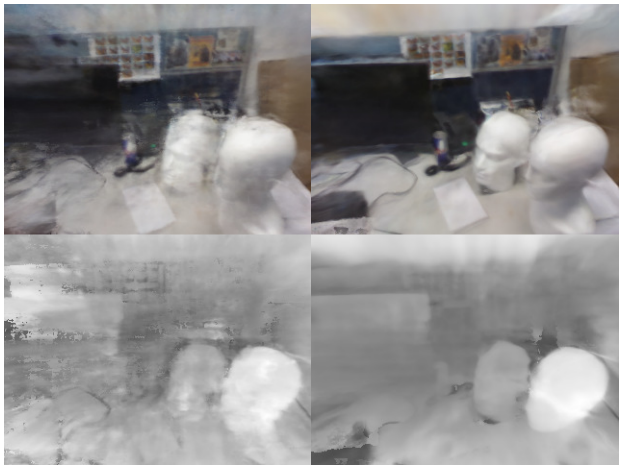
Table 2: We report the percentage of correctly re-localized frames below an error threshold of 5cm and percentage of re-localized frames below an error threshold $5°$ on the *Fern*, *Fortress*, *Horns*, *Room* scenes of LLFF dataset [33]

pose regression approach and the direct matching approach, resulting in the top performance.

4.4. Ablation Study

Effectiveness of Modified Positional Encoding In real-life datasets such as 7-Scenes, there are multiple sources to keep NeRF from rendering a high-quality, photorealistic view of the scene. Artifacts may be produced by letting the NeRF model learn from images with severe deformation (*e.g.*, from camera rolling shutter or deforming object) or motion blur from long exposure among frames. Moreover, training and testing in very different camera trajectories is a situation for NeRF likely to fail because it tries to generate the scene from unfamiliar volumetric rendering locations.

We build a toy example to demonstrate the phenomenon in the 7-Scenes dataset, and the effectiveness of our modification in NeRF's positional encoding. We randomly select a frame in Heads and sample a portion of training and validation data that lies within its frustum overlap threshold using an approach similar to Balntas *et al*. [1]. For this experiment, we set the frustum overlap threshold to be 0.85. We report the peak signal-to-noise ratio (PSNR) on this toy dataset for fixed full encoding ($m = 10$) in the original NeRF paper, fixed half encoding ($m = 5$), and the coarse-to-fine encoding schemes in Table 3. We show a qualitative comparison between the original NeRF embedding scheme and our modified coarse-to-fine scheme in Fig. 3. Overall, we find that using a coarse-to-fine positional encoding approach generally obtains a higher quality NeRF model throughout the 7-Scenes dataset in our experiments.



(a) Fixed P.E. [34] (b) Coarse-to-fine P.E.

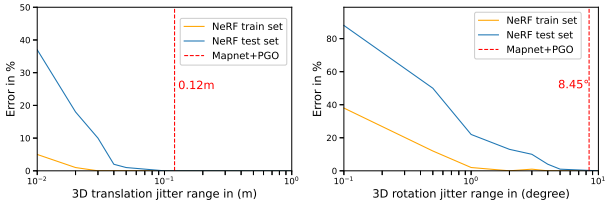
Figure 3: A visual comparison between (a) fixed positional encoding (P.E.) and (b) the coarse-to-fine P.E in Heads scene. Top: testset renderings from two NeRF models with different P.E. schemes. Bottom: disparity maps (inverse depth). Notice that NeRF with fixed P.E. produces stronger artifacts in this outward looking scene. Even though our encoding do not completely remove all artifacts, it recovers more structure and details than the original NeRF scheme. We provide more detailed discussion on why NeRF suffers from severe artifacts in Section 4.4.

Model	Full encoding	Half encoding	Coarse-to-fine
PSNR	16.64	17.16	**17.50**

Table 3: Comparison of different NeRF positional encoding scheme in our toy dataset (validation split).

Effectiveness of Direct Matching We investigate the effectiveness of direct matching for supervising the pose regression training. We first train a NeRF model using the Heads data. We randomly perturb the ground truth pose in different ranges and compute the photometric loss $\mathcal{L}_{ph\_perturb}$ with the ground truth image. We then count the percentage that $\mathcal{L}_{ph\_perturb} < \mathcal{L}_{ph\_GT}$, where $\mathcal{L}_{ph\_GT}$ denotes the photometric loss using the ground truth pose. We define such a percentage as the *error rate*. Ideally, views generated from perturbed poses should have higher photometric loss than the loss we get from using the ground truth poses. Thus the *error rate* indicates the chances that non-ideal cases would happen, which potentially damages the optimization of our pose regression.

Intuitively, the greater range we perturb the ground truth pose with, the more displacement in the appearance of rendered images will have and shall lead to a lower *error rate*. In this experiment, we jitter poses in the range between $\pm[0.01m, 1m]$ for 3D translation and the range between



(a) 3D translation jitter only (b) 3D rotation jitter only

Figure 4: We feed randomly perturbed poses to the NeRF model trained in Heads, and validate the robustness of our photometric loss $\mathcal{L}_{photo}$. By cumulatively counting the chances of $\mathcal{L}_{ph\_perturb} < \mathcal{L}_{ph\_GT}$, the *error rate* remains nearly 0 when the perturbation is smaller to the Mapnet+PGO reference threshold, for both translation and rotation. This indicates that the photometric loss from direct matching is effective to supervise the training of a pose regression network.

Scene	Geo. PoseNet [21]	PoseNet (ResNet34) [6]	PoseNet+logq (ResNet34) [6]	PoseNet+SE(3) (ResNet34)	PoseNet+SE(3) (MobileNetV2)
Backbone Error (Top-1/Top-5)	31.3%/11.1%	**26.7%/8.58%**	**26.7%/8.58%**	**26.7%/8.58%**	28.12%/9.71%
Average	0.23m, 8.12°	0.23m, 8.49°	0.22m, 8.07°	**0.21m**, 8.71°	**0.21m**, **7.84**°

Table 4: A comparison between our SE(3) PoseNet baseline and other quaternion-based baselines. Our direct SE(3) supervision offers competitive results with both backbones while removing the need for balancing rotation and translation terms.

$\pm[0.1°, 10°]$ for 3D rotation movement. For each of the selected scales, we randomly jitter 500 poses to estimate the expected *error rate*. As the results are shown in Fig. 4, both 3D translation and 3D rotation *error rates* drop close to 0 below the reference threshold of MapNet+PGO, which may explain why training with direct matching can obtain better performance overall.

Effectiveness of Regressing SO(3) We also compares our baseline pose regression model with prior baselines from PoseNet and MapNet in Table 4. We achieve on-par results by directly replacing the rotation representation from quaternion to SO(3) rotation representation. Our MobileNetV2 performs overall the best in terms of average results. We use identical training hyperparameters to train our baseline model for each scene, and our MobileNetV2 feature extractor is not the best regarding the ImageNet benchmark compared to prior baselines. Our baseline models' superior performance indicates that our rotation representation is just as effective as quaternion representation. A full table on scene specific performance is provided in the supplementary material.

Summary of Ablation We justify our design decisions to show how each component variation contributes to the relocalization performance in Table 5. First, replacing the SE(3) representation with separate quaternion rotation and translation position terms leads to lower accuracy due to the balancing requirement of the two terms during training. The most significant performance drop is when removing the coarse-to-fine training strategy on NeRF. It indicates that the NVS reconstruction quality does affect the overall relocalization accuracy. In addition, we observe that using full NeRF architecture to train our model can obtain slight accuracy improvement. However, this is at the cost of a much longer training time (i.e. 22hrs vs. 78hrs on Kitchen). We observe that the same phenomena hold valid when training with unlabeled data as well.

Method	7 Scenes
Direct-PN	0.20m, 7.26°
- SE(3)	0.21m, 7.58°
- Coarse-to-fine	0.22m, 7.91°
- Direct Matching	0.22m, 8.07°
Direct-PN + Full NeRF	**0.20m, 7.16**°

Table 5: A performance breakdown for each component in our method. The performance drops for when our modifications on SE(3), coarse-to-fine encoding, and the direct-matching module are removed. The performance improves slightly if a full-size NeRF [34] (with the hierarchical architecture, and a MLP with deeper and wider layers), but at the cost of a much longer training time.

5. Conclusion and Discussion

In this work, we show that one can use a differentiable renderer to improve pose regression performance. We present a relocalization pipeline that outperforms previous single-image APR methods on the 7-Scenes benchmark and achieves state-of-the-art performance on the LLFF dataset, with two main contributions. First, we joint a direct matching module with a pose regression network, offering superior performance while maintaining low inference cost. Second, we further boost our method's performance by applying a simple but effective semi-supervised training scheme to unlabeled data. To adapt with outward-looking relocalization datasets, which violates assumptions in NeRF, we employ a coarse-to-fine positional encoding strategy to improve rendering qualities.

One of the limitations of this work is that the effectiveness of our direct matching highly depends on the robustness of the NVS methods, which might fail for various reasons. For example, NeRF does not perform well in large-scale scenes, dynamic environments, or outdoor scenarios where auto-exposure fluctuates. The next challenge for us is to circumvent the assumptions made in NeRF so that our method is extensible to more challenging scenarios.

Acknowledgements We thank Kejie Li for his advice on experimental design and generous help to polish our paper. We also appreciate Henry Howard-Jenkins and Theo W. Costain for some great comments and discussions.

6. Supplementary

6.1. Implementation Details

Architectures Details The proposed pipeline includes a pose regression model for camera pose predictions and an NVS system for synthesis images. Specifically, we use a modified PoseNet model and a modified NeRF in our experiments. We summarize the details of the pose regression network architecture in Table 6. The architecture of our NeRF (Fig. 5) mainly follows the original implementation of Mildenhall *et al*. [34], except we apply a coarse-to-fine positional encoding $\gamma_\alpha(p)$ for both positions and directions, and we only use a coarse model with 128 samples along each ray. The entire implementation is written in PyTorch, and the NeRF code is built upon an open-sourced repository nerf-pytorch [62].

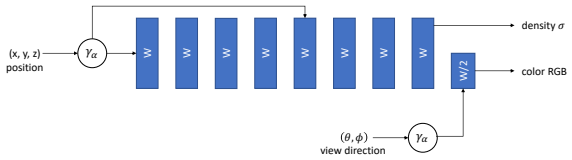


Figure 5: Our NeRF architecture. We adapt a coarse-to-fine encoding approach [39] for both positions and view directions to mitigate artifacts in NeRF reconstruction caused by outward-looking scenes and video distortions. We set $W = 128$ in our implementation.

Orthogonalize the Rotation Matrix As we have mentioned in the paper, the ground truth supervision $\mathcal{L}_{gt}$ is an approximation of the correct geometric loss. The regressed rotation matrix $\hat{\mathbf{R}}$ is not guaranteed to be in SO(3) manifold. To solve this, we apply a singular value decomposition (SVD) operation during testing:

$$\text{SVD}(\hat{\mathbf{R}}) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{5}$$

$$\hat{\mathbf{R}}_\mathbf{o} = \mathbf{U}\mathbf{V}^T, \tag{6}$$

where $\hat{\mathbf{R}}_\mathbf{o}$ denotes the orthogonalized rotation matrix.

Positional Encoding Strong artifacts occur in synthetic images if the NeRF is applied in the relocalization task without modification. This phenomenon appears because: a) NeRF is not designed for outward-looking scenes and b) photometric consistency is violated when auto-focus/exposure fluctuation and rolling shutter effect appear. As we addressed in the main paper, we mitigate this issue by adapting a coarse-to-fine positional encoding strategy $\gamma_\alpha(p)$ proposed by Nerfies [39], and illustrate this strategy

Input	Operator	t	m	n	s
$240 \times 320 \times 3$	conv2d	-	32	1	2
$120 \times 160 \times 32$	bottleneck	1	16	1	1
$120 \times 160 \times 16$	bottleneck	6	24	2	2
$60 \times 80 \times 24$	bottleneck	6	32	3	2
$30 \times 40 \times 32$	bottleneck	6	64	4	2
$15 \times 20 \times 64$	bottleneck	6	96	3	1
$15 \times 20 \times 96$	bottleneck	6	160	3	2
$8 \times 10 \times 160$	bottleneck	6	320	1	1
$8 \times 10 \times 320$	conv2d 1×1	-	1280	1	1
$8 \times 10 \times 1280$	avgpool	-	-	1	-
$1 \times 1 \times 1280$	fc	-	12	1	-

Table 6: Baseline pose regression network architecture of Direct-PoseNet, using an input image size $240 \times 320 \times 3$ as an example. The backbone is MobileNetV2 [45], with n repeated times for each operator and m output channels. s represents the stride and t represents the expansion factor.

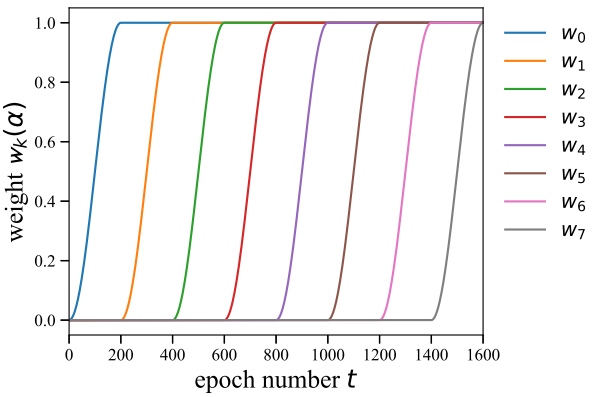


Figure 6: An example of w_k activation used in the main paper. The number of frequency band being activated increase as epoch iterations increase.

in Fig. 6. Specifically, an input signal p is encoded by

$$\begin{aligned}\gamma_\alpha(p) = [p, \ldots, w_k(\alpha_t)\sin(2^k\pi p),\\ w_k(\alpha_t)\cos(2^k\pi p), \ldots],\end{aligned} \tag{7}$$

where $0 \leq k \leq m-1, m \in \mathbb{N}$ and $w_k(\alpha_t)$ activates each band over epoch t, controlled by $\alpha_t = mt/N$. We denote N as the maximum epoch number in training and the weight $w_k(\alpha_t)$ is defined as:

$$w_k(\alpha_t) = \frac{(1 - \cos(\pi\,\text{clamp}(\alpha_t - k, 0, 1))}{2}. \tag{8}$$

6.2. Additional Ablation Study

More Results on the 7-Scenes Dataset We further compare our method with prior state-of-the-art methods on the

	without unlabeled data			with unlabeled data	
Model	PoseNet+logq [6]	MapNet [6]	Direct-PN	MapNet+PGO [6]	Direct-PN+U
Avg. Median	0.23m, 8.49°	0.21m, 7.77°	**0.20m, 7.26°**	0.18m, 6.55°	**0.16m, 5.17°**
Avg. Mean	0.28m, 10.43°	0.27m, 10.08°	**0.25m, 8.98°**	0.22m, 7.89°	**0.21m, 7.02°**

Table 7: A comparison of average median errors and average mean errors on the 7-Scenes dataset.

Scene	Geo. PoseNet [21]	PoseNet (ResNet34) [6]	PoseNet+logq (ResNet34) [6]	PoseNet+SE(3) (ResNet34)	PoseNet+SE(3) (MobileNetV2)
Backbone Error (Top-1/Top-5)	31.3%/11.1%	**26.7%/8.58%**	**26.7%/8.58%**	**26.7%/8.58%**	28.12%/9.71%
Chess	0.13m, 4.48°	**0.11m**, 4.24°	**0.11m**, 4.29°	**0.11m**, 4.53°	**0.11m**, **3.95°**
Fire	**0.27m**, 11.30°	0.29m, 11.68°	**0.27m**, 12.13°	0.28m, 11.65°	**0.27m**, **10.15°**
Heads	**0.17m**, 13.00°	0.20m, 13.11°	0.19m, **12.15°**	**0.17m**, 13.76°	**0.17m**, 13.30°
Office	0.19m, **5.55°**	0.19m, 6.40°	0.19m, 6.35°	0.18m, 5.92°	**0.17m**, 6.25°
Pumpkin	0.26m, 4.75°	0.23m, 5.77°	0.22m, 5.05°	**0.20m**, 6.11°	0.22m, **4.58°**
Kitchen	**0.23m**, 5.35°	0.27m, 5.81°	0.25m, **5.27°**	0.24m, 6.22°	0.24m, 5.47°
Stairs	0.35m, 12.40°	0.31m. 12.43°	0.30m, 11.29°	**0.29m**, 12.76°	0.30m, **11.20°**
Average	0.23m, 8.12°	0.23m, 8.49°	0.22m, 8.07°	**0.21m**, 8.71°	**0.21m**, **7.84°**

Table 8: A per scene based comparison between our SE(3) PoseNet baseline and other quaternion-based baselines, evaluated with median translation and rotation error on the 7-Scenes. Two columns to the right are results with our direct SE(3) supervision.

7-Scenes dataset (Table 7), showing our pipeline outperforms them in average median errors and average mean errors.

Table 8 shows the scene-specific comparison between SO(3) and quaternion-based representation. The quaternion-based results were provided by [21, 6], and their performances are confirmed based on their released code. The quaternion-based models were trained using geometric consistency loss [6], and the SE(3) models were trained using L2 loss without balancing translation and rotation terms.

About λs The reconstruction loss tends to have an order of magnitude larger value than the pose loss. In our paper, we didn't heavily tune the λs. We experimentally pull both losses into closer scales. Even with our default $\lambda_1 = 0.3, \lambda_2 = 0.7$ values, we observe that the weighted reconstruction loss is usually the dominant term of the combined loss, which proves the benefits of our architecture. Table 9 shows our experiments on 3 of 7-scenes with different λ settings. Although the result seems mixed, we argue both pose loss and photometric loss are important. For scenes with low texture or flat background, i.e., Lego scene from NeRF Synthetic dataset [34], pose loss ensures the regressed pose is regularized in relevant positions.

Dataset	Scene	$\lambda_1 = 0.1$ $\lambda_2 = 0.9$	$\lambda_1 = 0.3$ $\lambda_2 = 0.7$	$\lambda_1 = 0.5$ $\lambda_2 = 0.5$	$\lambda_1 = 0.7$ $\lambda_2 = 0.3$	$\lambda_1 = 0.9$ $\lambda_2 = 0.1$
7 Scenes	Heads	**0.17**, 13.08°	**0.17**, 13.1°	**0.17**, **12.87°**	**0.17**, 13.1°	**0.17**, 13.26°
7 Scenes	Fire	0.28, **8.48°**	**0.27**, 8.66°	**0.27**, 8.61°	**0.27**, 8.87°	**0.27**, 9.36°
7 Scenes	Pumpkin	**0.19**,**3.60°**	**0.19**,3.85°	**0.19**,3.77°	0.20,3.68°	**0.19**, 3.64°
NeRF Synthetic	Lego	0.167, 2.9°	**0.117**, **2.7°**	0.182, 4.7°	0.194, 5.1°	0.276, 5.8°

Table 9: Result of using different λ_1 and λ_2 values on Heads, Fire, and Pumpkin in 7-Scenes dataset (first 3 rows). We also tested in Lego scene (bottom row) on the NeRF synthesis datasets [34], which contains large areas of textureless background.

References

[1] V. Balntas, S. Li, and V. Prisacariu. Relocnet: Continuous metric learning relocalisation using neural nets. In *ECCV*, 2018. 1, 7

[2] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC - Differentiable RANSAC for Camera Localization. In *CVPR*, 2017. 1

[3] Eric Brachmann and Carsten Rother. Learning Less is More - 6D Camera Localization via 3D Surface Regression. In *CVPR*, 2018. 1

[4] Eric Brachmann and Carsten Rother. Neural- Guided RANSAC: Learning where to sample model hypotheses. In *ICCV*, 2019. 1

[5] Eric Brachmann and Carsten Rother. Visual camera relocalization from RGB and RGB-D images using DSAC. *arXiv*, 2020. 1

[6] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-Aware Learning of Maps for Camera Localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 4, 5, 6, 8, 10

[7] S.E. Chen and L. Williams. View interpolation for image synthesis. In *Computer Graphics Proceedings, Annual Conference Series*, 1993. 2

[8] Ondrej Chum and Jiri Matas. Optimal Randomized RANSAC. In *PAMI*, 2008. 1

[9] A. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Realtime single camera SLAM. In *IEEE Trans. Pattern Anal. Mach. Intell.*, 2007. 2

[10] J. Engel, V. Koltun, and D. Cremers. DSO: Direct sparse odometry. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017. 1, 2, 3, 5

[11] J. Engel, T. Schops, and D. Cremers. Semi-dense visual odometry for a monocular camera. In *In Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2013. 1, 2, 3

[12] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *CACM*, 1981. 1

[13] C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014. 2, 3

[14] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. SVO: Semi-direct visual odometry for monocular and multi-camera systems. In *IEEE Transactions on Robotics*, 2017. 2, 3

[15] Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time rgb-d camera relocalization. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, 2013. 5, 6

[16] Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICCV*, 2015. 4

[17] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996. 2

[18] M. Irani and P. Anandan. All about direct methods. In *Workshop Vis. Algorithms: Theory Pract.*, 1999. 2

[19] H. Jin, P. Favaro, and S. Soatto. Real-time 3-d motion and structure of point features: Front-end system for vision-based control and interaction. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2000. 2

[20] A. Kendall and R. Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *ICRA*, 2016. 1, 4

[21] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 4, 5, 6, 8, 10

[22] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *International Conference on Computer Vision*, 2015. 1, 2, 4, 5, 6

[23] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 4

[24] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *IEEE ACM Int. Symp. Mixed Augmented Reality*, 2007. 2

[25] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *IEEE Trans. Robot.*, 2015. 2

[26] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996. 2

[27] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *NeurIPS*, 2020. 2

[28] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. In *ACM Trans. Graph.*, 2019. 2

[29] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021. 2

[30] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu. Image-based localization using hourglass networks. In *ICCV Workshops*, 2017. 1, 2, 4, 5

[31] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[32] Moustafa Meshry, Dan B Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. In *In IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2

[33] 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. In *ACM TOG*, 2019. 5, 6

[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. 2, 3, 5, 7, 8, 9, 10

[35] T. Naseer and W. Burgard. Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017. 1

[36] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *Int. Conf. on Computer Vision (ICCV)*, 2011. 2, 3

[37] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *CVPR*, 2020. 3

[38] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[39] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Deformable Neural Radiance Fields. *arXiv preprint arXiv:2011.12948*, 2020. 2, 4, 9

[40] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019. 4

[41] Dai Peng, Zhang Yinda, Li Zhuwen, Liu Shuaicheng, and Zeng Bing. Neural point cloud rendering via multi-plane projection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[42] M. Pizzoli, C. Forster, and D. Scaramuzza. Remode: Probabilistic, monocular dense reconstruction in real time. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014. 2, 3

[43] N. Radwan, A. Valada, and W. Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. In *IEEE Robotics and Automation Letters*, 2018. 1, 4

[44] Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. USAC: A Universal Framework for Random Sample Consensus. In *PAMI*, 2013. 1

[45] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and LC. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 4, 9

[46] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *ICCV*, 2011. 1

[47] T. Sattler, B. Leibe, and L. Kobbelt. Improving image-based localization by active correspondence search. In *European conference on computer vision*, 2012. 1

[48] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. In *PAMI*, 2017. 1

[49] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[50] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[51] 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, 5, 6

[52] Y. Shvit and R. Ferens. Introduction to camera pose estimation with deep learning. *arXiv preprint arXiv: 1907.05272*, 2019. 1

[53] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. DeepVoxels: Learning persistent 3d feature embeddings. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[54] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structureaware neural scene representations. In *NeurIPS*, 2019. 3

[55] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, , and Akihiko Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. *CVPR*, 2018. 1

[56] Ayush Tewari, Christian Theobalt, Dan B Goldman, Eli Shechtman, Gordon Wetzstein, Jason Saragih, Jun-Yan Zhu, Justus Thies, Kalyan Sunkavalli, Maneesh Agrawala, Matthias Nießner, Michael Zollhöfer, Ohad Fried, Ricardo Martin Brualla, Rohit Kumar Pandey, Sean Fanello, Stephen Lombardi, Tomas Simon, and Vincent Sitzmann. State of the art on neural rendering. In *Computer Graphics Forum*, 2020. 2

[57] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. In *ACM Transactions on Graphics*, 2019. 2

[58] A. Valada, N. Radwan, and W. Burgard. Deep auxiliary learning for visual localization and odometry. In *ICRA*, 2018. 1, 4

[59] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-based localization using lstms for structured feature correlation. In *International Conference on Computer Vision*, 2017. 1, 2, 4, 5

[60] 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. 3

[61] J. Wu, L. Ma, and X. Hu. Delving Deeper into Convolutional Neural Networks for Camera Relocalization. In *ICRA*, 2017. 1, 2, 5

[62] Lin Yen-Chen. PyTorchNeRF: a PyTorch implementation of NeRF. https://github.com/yenchenlin/nerf-pytorch/, 2020. 9

[63] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Invert-

ing neural radiance fields for pose estimation. In *arxiv arXiv:2012.05877*, 2020. 3, 6

[64] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv:2010.07492*, 2020. 4

Statement of Authorship for the paper "Direct-PoseNet: Absolute Pose Regression with Photometric Consistency" in Chapter 3.

Paper title	Direct-PoseNet: Absolute Pose Regression with Photometric Consistency
Authors	**Shuai Chen**, Zirui Wang, Victor Adrian Prisacariu
Publication status	Published
Publication details	International Conference on 3D Vision (3DV), 2021.

Student Confirmation

Student name	Shuai Chen	
Contribution to the paper	First-author contribution: • Conception of research ideas • Design and implementation of models • Running of large-scale experiments • Writing and presentation of the paper	
Signature and Date		Apr. 23th 2025

Supervisor Confirmation

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

Supervisor name	Prof. Victor Adrian Prisacariu	
Supervisor comments	The description is accurate	
Signature and Date		Apr. 23th 2025

Chapter 4

Direct Feature-metric Matching with NeRF

This chapter explores the limitations of integrating direct matching formulations with APR methods, particularly in the presence of large photometric distortions. While Chapter 3 demonstrates that direct photometric matching offers a practical framework for training geometry-aware APR approaches, its effectiveness is constrained in environments with strong domain shifts, such as varying lighting, exposure, weather, or dynamic scene content. This is primarily because, in relocalisation, mapping and testing sequences are often captured at different times or under different conditions, leading to potentially significant appearance discrepancies. NeRF-based renderers, which mostly rely on appearance captured during mapping, often struggle to render accurately under these conditions due to their inability to generalise beyond observed illumination or exposure settings. To address the challenges, this chapter presents an enhanced APR training pipeline that extends the framework of the previous chapter by incorporating direct feature-metric matching and robust novel view synthesis.

The method consists of two main components: a histogram-assisted novel view

synthesiser (NeRF-Hist) and a feature-aware pose regression network DFNet. The NeRF-Hist model generates photometrically consistent synthetic views, compensating for changes in exposure, while DFNet simultaneously regresses camera poses and extracts robust features to bridge the domain gap between real and synthetic images. To ensure the effectiveness of feature matching, a contrastive learning scheme is applied to enforce the distinctiveness of the feature while maintaining sensitivity to transformation. Additionally, we propose Random View Synthesis (RVS), an online synthetic data generation strategy that efficiently augments training data, improving generalisation to unseen poses. Extensive experiments demonstrate that our method improves camera pose estimation in indoor and outdoor environments by as much as 56%. The rest of this chapter is based on our paper, *DFNet: Enhancing Absolute Pose Regression with Direct Feature Matching*, which was accepted at the *2022 European Conference on Computer Vision*.

DFNet: Enhance Absolute Pose Regression with Direct Feature Matching

Shuai Chen, Xinghui Li, Zirui Wang, and Victor A. Prisacariu

Active Vision Lab, University of Oxford

Abstract. We introduce a camera relocalization pipeline that combines absolute pose regression (APR) and direct feature matching. By incorporating exposure-adaptive novel view synthesis, our method successfully addresses photometric distortions in outdoor environments that existing photometric-based methods fail to handle. With domain-invariant feature matching, our solution improves pose regression accuracy using semi-supervised learning on unlabeled data. In particular, the pipeline consists of two components: Novel View Synthesizer and DFNet. The former synthesizes novel views compensating for changes in exposure and the latter regresses camera poses and extracts robust features that close the domain gap between real images and synthetic ones. Furthermore, we introduce an online synthetic data generation scheme. We show that these approaches effectively enhance camera pose estimation both in indoor and outdoor scenes. Hence, our method achieves a state-of-the-art accuracy by outperforming existing single-image APR methods by as much as 56%, comparable to 3D structure-based methods.[1]

Keywords: Absolute Pose Regression, Feature Matching, NeRF

1 Introduction

Estimating the position and orientation of cameras from images is essential in many applications, including virtual reality, augmented reality, and autonomous driving. While the problem can be approached via a geometric pipeline consisting of image retrieval, feature extraction and matching, and a robust Perspective-n-Points (PnP) algorithm, many challenges remain, such as invariance to appearance or the selection of the best set of method hyperparameters.

Learning-based methods have been used in traditional pipelines to improve robustness and accuracy, e.g. by generating neural network (NN)-based feature descriptors [7,16,17,25,26], combining feature extraction and matching into one network [30], or incorporating differentiable outlier filtering modules [1,2,3]. Although deep 3D-based solutions have demonstrated favorable results, many prerequisites often remain, such as the need for an accurate 3D model of the scene and manual hyperparameter tuning of the remaining classical components.

The alternative end-to-end NN-based approach, termed absolute pose regression (APR), directly regresses the absolute pose of the camera from input images

[1] The code is available in https://code.active.vision.

[14] without requiring prior knowledge about the 3D structure of the neighboring environment. Compared with deep 3D-based methods, APR methods can achieve at least one magnitude faster running speeds at the cost of inferior accuracy and longer training time. Although follow-up works such as MapNet [4] and Kendall *et al.* [13] attempt to improve APR methods by adding various constraints such as relative pose and scene geometry reprojection, a noticeable gap remains between APR and 3D-based methods.

Recently, Direct-PN [5] achieved state-of-the-art (SOTA) accuracy in indoor localization tasks among existing single-frame APR methods. As well as being supervised by ground-truth poses, the network directly matches the input image and a NeRF-rendered image at the predicted pose. However, it has two major limitations: (a) direct matching is very sensitive to photometric inconsistency, as images with different exposures could produce a high photometric error even from the same camera pose, which reduces the viability of photometric direct matching in environments with large photometric distortions, such as outdoor scenes; (b) there is a domain gap between real and rendered images caused by poor rendering quality or changes in content and appearance of the query scene.

In order to address these limitations, we propose a novel relocalization pipeline that combines APR and direct feature matching. First, we introduce a histogram-assisted variant of NeRF, which learns to control synthetic appearance via histograms of luminance information. This significantly reduces the gap between real and synthetic image appearance. Second, we propose a network *DFNet* that extracts domain invariant features and regresses camera poses, trained using a contrastive loss with a customized mining method. Matching these features instead of direct pixels colors boosts the performance of the direct dense matching further. Third, we improve generalizability by (i) applying a cheap Random View Synthesis (RVS) strategy to efficiently generate a synthetic training set by rendering novel views from randomly generated pseudo training poses and (ii) allow the use of unlabeled data. We show that our method outperforms existing single-frame APR methods by as much as 56% on both indoor 7-Scenes and outdoor Cambridge datasets. We summarize our main contributions as follows:

2 Related Work

Absolute Pose Regression Absolute pose regression aims to directly regress the 6-DOF camera pose from an image using Convolutional Neural Networks. The first practice in this area is introduced by PoseNet [14], which is a GoogLeNet-backbone network appended with an MLP regressor. Successors of PoseNet propose several variations in network architectures, such as adding LSTM layers [36], adapting an encoder-decoder backbone [19], splitting the network into position and orientation branches [38], or incorporating attentions using transformers [29,28]. Other methods propose different strategies to train APR. Bayesian PoseNet [15] inserts Monte Carlo dropout to a Bayesian CNN that estimates pose with uncertainty. Kendall *et al.* [13] proposes to balance the translation and rotation loss at training using learnable weights and reprojection error. Map-

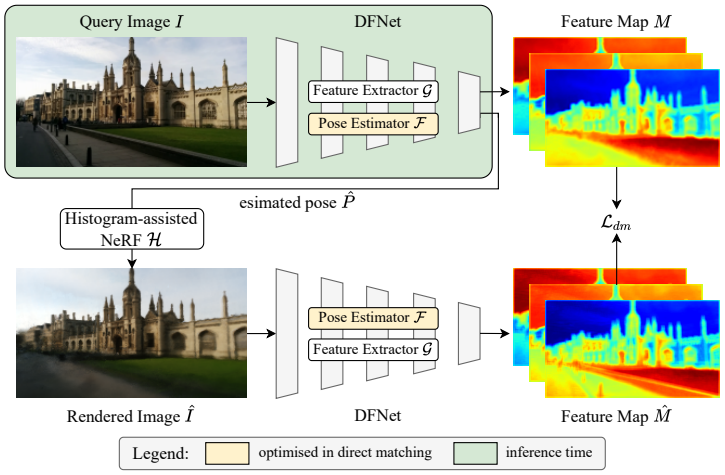


Fig. 1. Overview of the direct feature matching pipeline. Given an input image I, a pose regressor $\mathcal{F}$ estimates a camera pose $\hat{P}$, from which a luminance prior NVS system $\mathcal{H}$ renders a synthetic image $\hat{I}$. Domain invariant features of M and $\hat{M}$ are extracted using a feature extractor $\mathcal{G}$, supplying a feature-metric direct matching signal $\mathcal{L}_{dm}$ to optimize the pose regressor.

Net [4] trains the network using both absolute pose loss and relative pose loss but can infer in a single-frame manner. Direct-PoseNet (Direct-PN) [5] adapts additional photometric loss by comparing the query image with NeRF synthesis on the predicted pose.

Semi-supervised Learning in APR Several APR methods explore semi-supervised learning with additional images without ground-truth pose annotation to improve pose regression performance. To the best of our knowledge, MapNet+ [4] and MapNet+PGO [4] are the pioneers to train APR on unlabeled video sequences using external VO algorithms [8,9]. Direct-PN+ [5] finetune on unlabeled data from arbitrary viewpoints solely based on its direct matching formulation. While the direct matching idea from Direct-PN+ inspires our proposed method, we focus on training in the feature space. Our solution can scale to scenes with large photometric distortion, where the previous method fails.

Novel View Synthesis in APR Novel View Synthesis (NVS) can be beneficial to the visual relocalization task. For example, NVS can expand training space by generating extra synthetic data. Purkait *et al.* [24] propose a method to generate realistic synthetic training data for pose regression leveraging the 3D map and feature correspondences. LENS [22] deploys a NeRF-W [18] model to sample the scene boundaries and synthesize virtual views with uniformly generated virtual camera poses. However, Purkait *et al.* rely on a pre-computed reconstructed 3D map. LENS is limited by its costly offline computation efficiency and the lack of compensation to the domain gap between synthetic and real images, i.e.,

dynamic objects or artifacts. Another direction is to embed NVS into the pose estimation process. InLoc [32] verifies the predicted pose with view synthesis. Ng et al. [23] combine a multi-view stereo (MVS) model with a relative pose regressor (RPR). iNeRF [39], Wang *et al.* [37], and Direct-PN [5] utilize an inverted NeRF to optimize the camera pose. Our paper is the first to incorporate both strategies yet have major differences from the above methods. 1) we introduce an NVS method that can adapt to real exposure change in view synthesis. 2) we address the domain adaptation problem between the actual camera footage with synthetic images. 3) our synthetic data generation strategy is comparatively less constrained and can be deployed efficiently in online training.

3 Method

We illustrate our proposed direct feature matching pipeline in Fig. 1, which contains two primary components: 1) the DFNet network, which, given an input image I, uses a pose estimator $\mathcal{F}$ to predict a 6-DoF camera pose and a feature extractor $\mathcal{G}$ to compute a feature map M, and 2) a histogram-assisted NeRF $\mathcal{H}$, which compensates for high exposure fluctuation by providing luminance control when rendering a novel view given an arbitrary pose.

Training the direct feature matching pipeline can be split into two stages, (i) DFNet and the histogram-assisted NeRF, and (ii) direct feature matching. In stage one, we train the NVS module $\mathcal{H}$ like a standard NeRF, and the DFNet with a loss term $\mathcal{L}_{DFNet}$ in Eq. (5). In stage two, fixing the histogram-assisted NeRF and the feature extractor $\mathcal{G}$, we further optimize the main pose estimation module $\mathcal{F}$ via a direct feature matching signal between feature maps extracted from the real image and its synthetic counterpart $\hat{I}$, which is rendered from the predicted pose $\hat{P}$ of image I via the NVS module $\mathcal{H}$. At test time, only the pose estimator $\mathcal{F}$ is required given the query image, which ensures a rapid inference.

This section is organized as follows: the DFNet pipeline is detailed in Section 3.1, followed by a showcase of our histogram-assisted NeRF $\mathcal{H}$ in Section 3.2. To further boost the pose estimation accuracy, an efficient Random View Synthesis (RVS) training strategy is introduced in 3.3.

3.1 Direct Feature Matching For Pose Estimation

This section aims to introduce: 1) the design of our main network DFNet, 2) the direct feature matching formulation that boosts pose estimation performance in a semi-supervised training manner, and 3) the contrastive-training scheme that closes the domain gap between real images and synthetic images.

DFNet Structure The DFNet in our pipeline consists of two networks, a pose estimator $\mathcal{F}$ and a feature extractor $\mathcal{G}$. The pose estimator $\mathcal{F}$ in our DFNet is similar to an ordinary PoseNet, which predicts a 6-DoF camera pose $\hat{P} = \mathcal{F}(I)$ for an input image I, and can be supervised by an L_1 or L_2 loss between the pose estimation $\hat{P}$ and its ground truth pose P.

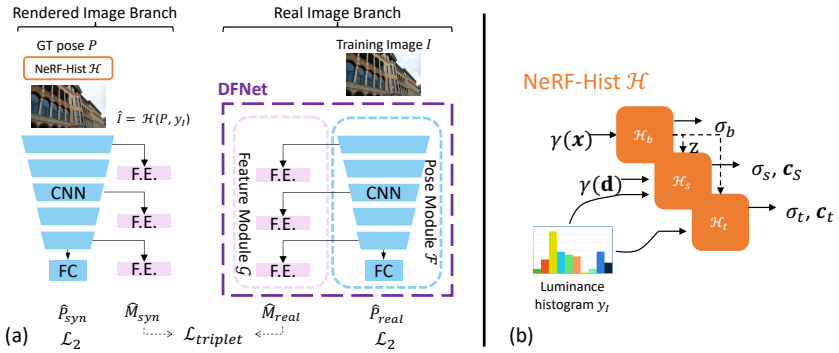


Fig. 2. (a) The training scheme for DFNet to close the domain gap between real images and rendered images. (b) The histogram-assisted NeRF architecture.

The feature extractor $\mathcal{G}$ in our DFNet takes as input feature maps extracted from various convolutional blocks in the pose estimator and pushes them through a few convolutional blocks, producing the final feature maps $M = \mathcal{G}(I)$, which are the key ingredients during feature-metric direct matching.

Two key properties of the feature extractor $\mathcal{G}$ that we seek to learn are 1) domain invariance, i.e., being invariant to the domain of real images and the domain of synthetic images and 2) transformation sensitive, i.e., being sensitive to the image difference that is caused by geometry transformations. With these properties learned, our feature extractor can extract domain-invariant features during feature-metric direct matching while preserving geometry-sensitive information for pose learning. We detail the way to train the DFNet in the *Closing the Domain Gap* section.

Direct Feature Matching Direct matching in APR was first introduced by Direct-PN [5], which minimizes the photometric difference between a real image I and a synthetic image $\hat{I}$ rendered from the estimated pose $\hat{P}$ of the real image I. Ideally, if the predicted pose $\hat{P}$ is close to its ground truth pose P, and the novel view renderer produces realistic images, the rendered image $\hat{I}$ should be indistinguishable from the real image.

In practice, we found the photometric-based supervision signal could be noisy in direct matching, when part of scene content changes. For example, random cars and pedestrians may appear through time or the NeRF rendering quality is imperfect. Therefore, we propose to measure the distance between images in feature space instead of in photometric space, given that the deep features are usually more robust to appearance changes and imperfect renderings.

Specifically, for an input image I and its pose estimation $\hat{P} = \mathcal{F}(I)$, a synthetic image $\hat{I} = \mathcal{H}(\hat{P}, \mathbf{y}_I)$ can be rendered using the pose estimation $\hat{P}$ and the histogram embedding $\mathbf{y}_I$ of the input image I. We then extract the feature

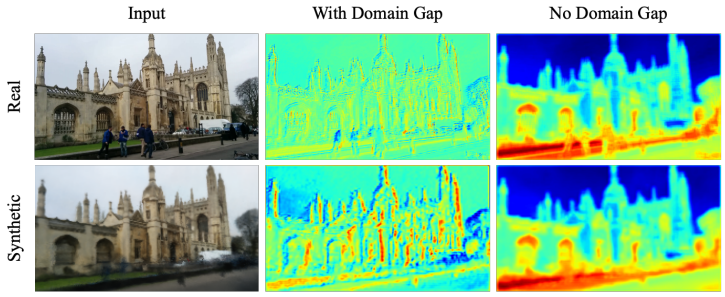


Fig. 3. A visual comparison of features before and after closing the domain gap. Ideally, a robust feature extractor shall produce indistinguishable features between real and rendered images from the same pose. Column 2/Column 3 are features trained without/with using our proposed $\mathcal{L}_{triplet}$ loss, where our method can effectively produce similar features across two domains.

map $M \in \mathbb{R}^{H_M \times W_M \times C_M}$ and $\tilde{M} \in \mathbb{R}^{H_M \times W_M \times C_M}$ for image I and $\hat{I}$ respectively, where H_M and W_M are the spatial dimensions and C_M is the channel dimension of the feature maps. To measure the difference between two feature maps, we compute a cosine similarity between feature $m_i \in \mathbb{R}^{C_M}$ and $\tilde{m}_i \in \mathbb{R}^{C_M}$ for each feature location i:

$$\cos(m_i, \tilde{m}_i) = \frac{m_i \cdot \tilde{m}_i}{\|m_i\|_2 \cdot \|\tilde{m}_i\|_2}. \tag{1}$$

By minimizing the feature-metric direct matching loss $\mathcal{L}_{dm} = \sum_i (1 - \cos(m_i, \tilde{m}_i))$, the pose estimator $\mathcal{F}$ can be trained in a semi-supervised manner (note no ground truth label required for the input image I).

Our direct feature matching may optionally follow the procedure of semi-supervised training proposed by MapNet+ [4] to improve pose estimation with unlabeled sequences captured in the same scene. Unlike [4], which requires sequential frames to enforce a relative geometric constraint using a VO algorithm, our feature-matching can be trained by images from arbitrary viewpoints without ground truth pose annotation. Our method can be used at train time with a batch of unlabeled images, or as a pose refiner for a single test image. In the latter case, our direct matching can also be regarded as a post-processing module. During the training stage, only the weights of the pose estimator will be updated, whereas the feature extractor part remains frozen to back-propagation.

Closing the Domain Gap We notice that synthetic images from NeRF are imperfect due to rendering artifacts or lack of adaption of the dynamic content of the scene, which leads to a domain gap between render and real images. This domain gap poses difficulties to our feature extractor (Fig. 3), which we expect to produce features far away if two views are from different poses and to produce similar features between a rendered view and a real image from the same pose.

Intuitively, we could simply enforce the feature extractor to produce similar features for a rendered image $\hat{I}$ and a real image I via a distance function $d(\cdot)$ during training. However, this approach leads to model collapse [6], which motivates us to explore the original triplet loss:

$$\mathcal{L}_{triplet}^{ori} = \max\left\{d(M_{real}^{P}, M_{syn}^{P}) - d(M_{real}^{P}, M_{syn}^{\bar{P}}) + \text{margin}, 0\right\}, \tag{2}$$

where M_{real}^{P} and M_{syn}^{P}, the feature maps of a real image and a synthetic image at pose P, compose a positive pair, and $M_{syn}^{\bar{P}}$ is a feature map of a synthetic image rendered at an arbitrary pose $\bar{P}$ other than the pose P.

With a closer look at the task of feature-metric direct matching, we implement a customized in-triplet mining which explores the minimum distances among negative pairs:

$$\mathcal{L}_{triplet} = \max\left\{d(M_{real}^{P}, M_{syn}^{P}) - q_{\ominus} + \text{margin}, 0\right\}, \tag{3}$$

where the positive pair is as same as Eq. (2) and $q_{\ominus}$ is the minimum distance between four negative pairs:

$$q_{\ominus} = \min\left\{d(M_{real}^{P}, M_{real}^{\bar{P}}), d(M_{real}^{P}, M_{syn}^{\bar{P}}), d(M_{syn}^{P}, M_{real}^{\bar{P}}), d(M_{syn}^{P}, M_{syn}^{\bar{P}})\right\}, \tag{4}$$

which essentially takes the hardest negative pair among all matching pairs between synthetic images and real images that are in different camera poses. The margin value is set to 1.0 in our implementation. Since finding the minimum of negative pairs is non-differentiable, we implement the in-triplet mining as a prior step before $\mathcal{L}_{triplet}$ is computed.

Overall, to train the pose estimator and to obtain domain invariant and transformation sensitive property, we adapt a siamese-style training scheme as illustrated in Fig. 2a. Given an input image I and its ground truth pose P, a synthetic image $\hat{I}$ can be rendered via the NVS module $\mathcal{H}$ (assumed pre-trained) using the ground truth pose P. We then present both the real image I and the synthetic image $\hat{I}$ to the pose estimator and the feature extractor, resulting in pose estimations $\hat{P}_{real}$ and $\hat{P}_{syn}$ and feature maps M_{real} and M_{syn} for the real image I and synthetic image $\hat{I}$, respectively. The training then is supervised via a combined loss function

$$\mathcal{L}_{DFNet} = \mathcal{L}_{triplet} + \mathcal{L}_{RVS} + \frac{1}{2}(\|P - \hat{P}_{real}\|_2 + \|P - \hat{P}_{syn}\|_2), \tag{5}$$

where $\|\cdot\|$ denotes a L_2 loss and $\mathcal{L}_{RVS}$ is a supervision signal from our RVS training strategy, which we explain in Section 3.3.

3.2 Histogram-assisted NeRF

The DFNet pipeline relies on an NVS module that renders a synthetic image from which we extract a feature map and compare it with a real image. Theoretically,

Fig. 4. Typically NeRF only renders views that reflect the appearance of its training sequences, as shown by NeRF-W's synthetic view (b). However, in relocalization tasks, the query set may have different appearances or exposures to the train set. The proposed histogram-assisted NeRF (c) can render a more accurate appearance to the unseen query set (a) in both quantitative (PSNR) and visual comparisons. We refer to the supplementary for more examples.

while the NVS module in our pipeline can be in any form as long as it provides high-quality novel view renderings, in practice, we found that due to the presence of auto exposure during image capturing, it is necessary to have a renderer that can render images in a compensated exposure condition. Although employing direct matching in feature space could mediate the exposure issue to some extent, we find decoupling the exposure issue from the domain adaption issue leads to better pose estimation results.

One off-the-shelf option is a recent work NeRF-W [18], which offers the ability to control rendered appearance via an appearance embedding that is based on frame indices. However, in the context of direct matching, since we aim to compare a real image with its synthetic version, we desire a more fine-grained exposure control to render an image that matches the exposure condition of the real image, as illustrated in Fig. 4.

To this end, we propose a novel view renderer histogram-assisted NeRF (Fig. 2b) which renders an image $\hat{I} = \mathcal{H}(P, \mathbf{y}_I)$ that matches the exposure level of a query real image I via a histogram embedding $\mathbf{y}_I$ of the query image I at an arbitrary camera pose P. Specifically, our NeRF contains 3 components:

1. A base network $\mathcal{H}_b$ that provides a density estimation σ_b and a hidden state $\mathbf{z}$ for a coarse estimation: $[\sigma_b, \mathbf{z}] = \mathcal{H}_b(\gamma(\mathbf{x}))$.
2. A static network $\mathcal{H}_s$ to model density σ_s and radiance $\mathbf{c}_s$ for static structure and appearance: $[\sigma_s, \mathbf{c}_s] = \mathcal{H}_s(\mathbf{z}, \gamma(\mathbf{d}), \mathbf{y}_I)$.
3. A transient network $\mathcal{H}_t$ to model density σ_t, radiance $\mathbf{c}_t$ and an uncertainty estimation β for dynamic objects: $[\sigma_t, \mathbf{c}_t, \beta] = \mathcal{H}_t(\mathbf{z}, \mathbf{y}_I)$.

As for the input, $\mathbf{x}$ is a 3D point and $\mathbf{d}$ is a view angle that observes the 3D point, with both of them encoded by a positional encoding [10,35,20] operator $\gamma(\cdot)$ before injecting to each network.

During training, the coarse density estimation from the base network $\mathcal{H}_b$ provides a distribution where the other two networks could sample more 3D points near non-empty space accordingly. Both the static and the transient network are conditioned on a histogram-based embedding $\mathbf{y}_I \in \mathbb{R}^{C_y}$, which is mapped from a N_b bins histogram. The histogram is computed on the luma channel Y of a target

image in YUV space. We found this approach works well in a direct matching context, not only in feature-metric space but also in photometric space.

We adopt a similar network structure and volumetric rendering method as in NeRF-W[18], to which we refer readers for more details.

3.3 Random View Synthesis

During the training of DFNet, we can generate training data by synthesis more views from randomly perturbed training poses. We refer this process as Random View Synthesis (RVS), and we use this data generation strategy to help the DFNet to better generalize to unseen views.

Specifically, given a training pose P, a perturbed pose P' can be generated around the training pose with a random translation noise of ψ meters and random rotation noise of ϕ degrees. A synthetic image $I' = \mathcal{H}(P', \mathbf{y}_{I_{nn}})$ is then rendered via histogram-assisted NeRF $\mathcal{H}$, with $\mathbf{y}_{I_{nn}}$ being the histogram embedding of the training image with the nearest training pose. The synthetic pose-image pair (P', I') is used as a training sample for the pose estimator to provide an additional supervision signal $\mathcal{L}_{RVS} = \|P' - \hat{P}'\|_2$, where $\hat{P}' = \mathcal{F}(I')$ is the pose estimation of the rendered image.

A key advantage of our method is efficiency in comparison with prior training sample generation methods. For example, LENS [22] generates high-resolution synthetic data with a maximum of 40s/image and requires complicated parameter settings in finding candidate poses within scene volumes. In contrast, our RVS is a lightweight strategy that seamlessly fits our DFNet training at a much cheaper cost (12.2 fps) and with fewer constraints in pose generation while being able to reach similar performance. We refer to Section 4.5 for more discussion.

4 Experiments

4.1 Implementation

We introduce the implementation details for histogram-assisted NeRF, DFNet, and direct feature matching. We also provide more details in the supplementary.

NeRF Our histogram-assisted NeRF model is trained with a re-aligned and re-centered pose in SE(3), similar to Mildenhall *et al.* [20]. The image histogram bin size is set to $N_b = 10$ and embedded with a vector dimension of 50 for the static model and 20 for the transient model. We train the model with a learning rate of 5×10^{-4} and an exponential decay of 5×10^{-4} for 600 epochs.

DFNet Our DFNet adapts an ImageNet pre-trained VGG-16 [31] as the backbone, and an Adam optimizer with a learning rate of 1×10^{-4} is applied during training. For feature extraction, we extract $L = 3$ feature maps from the end of the encoder's first, third, and fifth blocks before pooling layers. All final feature outputs are upscaled to the same size as the input image $H \times W$ with bilinear upsampling. For pose regression, we regresses the SE(3) camera pose with a fully connected layer. A singular value decomposition (SVD) is applied to ensure the rotation component of $\hat{P}$ is normalized [5].

Table 1. Pose regression results on 7-Scenes dataset. We compare DFNet and DFNet$_{dm}$ (DFNet with feature-metric direct matching) with prior single-frame APR methods and unlabeled training methods, in median translation error (m) and rotation error (°). Note that MapNet+ and MapNet+PGO are sequential methods with unlabeled training. Numbers in **bold** represent the best performance.

	Methods	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Average
1-frame APR	PoseNet(PN)[14]	0.32/8.12	0.47/14.4	0.29/12.0	0.48/7.68	0.47/8.42	0.59/8.64	0.47/13.8	0.44/10.4
	PN Learn σ^2[13]	0.14/4.50	0.27/11.8	0.18/12.1	0.20/5.77	0.25/4.82	0.24/5.52	0.37/10.6	0.24/7.87
	geo. PN[13]	0.13/4.48	0.27/11.3	0.17/13.0	0.19/5.55	0.26/4.75	0.23/5.35	0.35/12.4	0.23/8.12
	LSTM PN[36]	0.24/5.77	0.34/11.9	0.21/13.7	0.30/8.08	0.33/7.00	0.37/8.83	0.40/13.7	0.31/9.85
	Hourglass PN[19]	0.15/6.17	0.27/10.8	0.19/11.6	0.21/8.48	0.25/7.0	0.27/10.2	0.29/12.5	0.23/9.53
	BranchNet[38]	0.18/5.17	0.34/8.99	0.20/14.2	0.30/7.05	0.27/5.10	0.33/7.40	0.38/10.3	0.29/8.30
	MapNet[4]	0.08/3.25	0.27/11.7	0.18/13.3	0.17/5.15	0.22/4.02	0.23/**4.93**	0.30/12.1	0.21/7.77
	Direct-PN[5]	0.10/3.52	0.27/8.66	0.17/13.1	0.16/5.96	0.19/3.85	0.22/5.13	0.32/10.6	0.20/7.26
	TransPoseNet[29]	0.08/5.68	0.24/10.6	0.13/12.7	0.17/6.34	0.17/5.6	0.19/6.75	0.30/7.02	0.18/7.78
	MS-Transformer[28]	0.11/4.66	0.24/9.60	0.14/12.2	0.17/5.66	0.18/4.44	**0.17**/5.94	0.17/5.94	0.18/7.28
	DFNet (ours)	**0.05/1.88**	**0.17/6.45**	**0.06/3.63**	**0.08/2.48**	**0.10/2.78**	0.22/5.45	**0.16/3.29**	**0.12/3.71**
Unlabel Data	MapNet$_{+(seq.)}$[4]	0.10/3.17	0.20/9.04	0.13/11.1	0.18/5.38	0.19/3.92	0.20/5.01	0.30/13.4	0.19/7.29
	MapNet$_{+PGO(seq.)}$[4]	0.09/3.24	0.20/9.29	0.12/8.45	0.19/5.42	0.19/3.96	0.20/4.94	0.27/10.6	0.18/6.55
	Direct-PN+U[5]	0.09/2.77	0.16/4.87	0.10/6.64	0.17/5.04	0.19/3.59	0.19/4.79	0.24/8.52	0.16/5.17
	DFNet$_{dm}$ (ours)	**0.04/1.48**	**0.04/2.16**	**0.03/1.82**	**0.07/2.01**	**0.09/2.26**	**0.09/2.42**	**0.14/3.31**	**0.07/2.21**

Direct Feature Matching To validate our feature-metric direct matching formulation, we follow the same procedure from MapNet+ [4] and Direct-PN+U [5], which use a portion of validation images without the ground truth poses for finetuning. When finetuning DFNet, we optimize the pose regression module $\mathcal{F}$ solely based on the direct feature matching loss $\mathcal{L}_{dm}$. We set the batch size to 1 and the learning rate to 1×10^{-5}. For naming simplicity, we named our model trained with direct feature matching as DFNet$_{dm}$.

4.2 Evaluation on the 7-Scenes Dataset

We evaluate our method on an indoor camera localization dataset 7-Scenes [11,30]. The dataset consists of seven indoor scenes scaled from $1m^3$ to $18m^3$. Each scene contains 1000 to 7000 training sets and 1000 to 5000 validation sets. Both histogram-assisted NeRF and DFNet use subsampled training data with a spacing window $d = 5$ for scenes containing ≤ 2000 frames and $d = 10$ otherwise. RVS poses are sampled on the training pose, and the DFNet parameters are $t_\psi = 0.2m$, $r_\phi = 10°$, and $d_{max} = 0.2m$. For fair comparison to other unlabeled training methods such as MapNet+ and Direct-PN, we finetune our DFNet$_{dm}$ using the same amount of unlabeled samples, which is 1/5 or 1/10 of the sequences based on the spacing window above to ensure our method is not overfitting to the entire test sequences.

We compared our method quantitatively with prior single-frame APR methods and unlabeled training APR methods in Table 1. The results show that both our DFNet and DFNet$_{dm}$ obtain superior accuracy, and DFNet$_{dm}$ achieves 56% and 57% improvement over averaged median translation and rotation errors compared to prior SOTA performance.

Table 2. Single-frame APR results on Cambridge dataset. We report the median position and orientation errors in $m/^\circ$ and the respective rankings over scene average as in [29,28]. The best results is highlighted in **bold**. For fair comparisons, we omit prior APR methods which did not publish results in Cambridge.

Methods	Kings	Hospital	Shop	Church	Average	Ranks	Final Rank
PoseNet(PN)[14]	1.66/4.86	2.62/4.90	1.41/7.18	2.45/7.96	2.04/6.23	9/9	9
PN Learn σ^2[13]	0.99/1.06	2.17/2.94	1.05/3.97	1.49/3.43	1.43/2.85	6/3	5
geo. PN[13]	0.88/1.04	3.20/3.29	0.88/3.78	1.57/3.32	1.63/2.86	7/4	6
LSTM PN[36]	0.99/3.65	1.51/4.29	1.18/7.44	1.52/6.68	1.30/5.51	5/8	7
MapNet[4]	1.07/1.89	1.94/3.91	1.49/4.22	2.00/4.53	1.63/3.64	7/7	8
TransPoseNet[29]	0.60/2.43	1.45/3.08	0.55/3.49	1.09/4.94	0.91/3.50	2/6	3
MS-Transformer[28]	0.83/1.47	1.81/2.39	0.86/3.07	1.62/3.99	1.28/2.73	4/2	2
DFNet (ours)	0.73/2.37	2.00/2.98	0.67/2.21	1.37/4.03	1.19/2.90	3/5	3
DFNet$_{dm}$ (ours)	**0.43/0.87**	**0.46/0.87**	**0.16/0.59**	**0.50/1.49**	**0.39/0.96**	**1/1**	**1**

Table 3. Comparison between our method and sequential-based APR methods and 3D structure-based methods.

	3D	Seq. APR				1-frame
Methods	AS[27]	MapNet +PGO[4]	CoordiNet[21]	CoordiNet +Lens[22]	VLocNet[34]	DFNet$_{dm}$
Chess	0.04/2.0	0.09/3.24	0.14/6.7	0.03/1.3	0.04/1.71	0.04/1.48
Fire	0.03/1.5	0.20/9.29	0.27/11.6	0.10/3.7	0.04/5.34	0.04/2.16
Heads	0.02/1.5	0.12/8.45	0.13/13.6	0.07/5.8	0.05/6.65	0.03/1.82
Office	0.09/3.6	0.19/5.42	0.21/8.6	0.07/1.9	0.04/1.95	0.07/2.01
Pumpkin	0.08/3.1	0.19/3.96	0.25/7.2	0.08/2.2	0.04/2.28	0.09/2.26
Kitchen	0.07/3.4	0.20/4.94	0.26/7.5	0.09/2.2	0.04/2.21	0.09/2.42
Stairs	0.03/2.2	0.27/10.6	0.28/12.9	0.14/3.6	0.10/6.48	0.14/3.31
Average	0.05/2.5	0.18/6.55	0.22/9.7	0.08/3.0	**0.05**/3.80	0.07/**2.21**
Kings	0.42/0.6	-	0.70/2.92	0.33/0.5	0.84/1.42	0.43/0.87
Hospital	0.44/1.0	-	0.97/2.08	0.44/0.9	1.08/2.41	0.46/0.87
Shop	0.12/0.4	-	0.73/4.69	0.27/1.6	0.59/3.53	0.16/0.59
Church	0.19/0.5	-	1.32/3.56	0.53/1.6	0.63/3.91	0.50/1.49
Average	0.29/0.63	-	0.92/2.58	**0.39**/1.15	0.78/2.82	**0.39/0.96**

4.3 Evaluation on Cambridge Dataset

We further compare our approach on four outdoor scenes from the Cambridge Landmarks [14] dataset, scaling from $875m^2$ to $5600m^2$. Each scene contains from 200+ to 1500 training samples. Our models are trained with 50% of training data, and DFNet's RVS are $t_\psi = 3m$, $r_\phi = 7.5^\circ$, and $d_{max} = 1m$. For finetuning DFNet$_{dm}$ with unlabeled data, we use 50% of the unlabeled validation sequence since fewer validation sets are available than 7-Scenes. Table 2 shows a comparison between our approach and prior single-frame APR methods, which omits prior APR methods that did not report results in Cambridge. We observe that our DFNet$_{dm}$ outperforms other methods significantly (60%+ in scene average), which further proves the effectiveness of our approach.

Table 4. (a) The effect of various level of features on DFNet$_{dm}$ result. Letter F, M, and C denote features extracted from fine, middle, and coarse levels in DFNet.
(b) Ablation on DFNet (upper part) and histogram-assisted NeRF in photometric direct matching (lower part). DFM denotes Direct Feature Matching.

(a) Featrue level vs. pose error

Feature Level	DFNet$_{dm}$ (ShopFacade)
F	**0.15m**, **0.64°**
F+M	0.19m, 0.77°
F+M+C	0.20m, 0.77°

(b) Ablation

Method	Shop Facade
DFNet w/ $\mathcal{L}_{triplet}^{ori}$	1.49m/5.80°
+RVS	0.86m/4.05°
+$\mathcal{L}_{triplet}$	0.72m/2.58°
+DFM (NeRF-W)	0.43m/1.62°
+DFM (NeRF-Hist)	**0.15m/0.65°**
Direct-PN	1.10m/4.25°
Direct-PN+U	1.41m/6.97°
+ NeRF-Hist	0.72m/3.39°

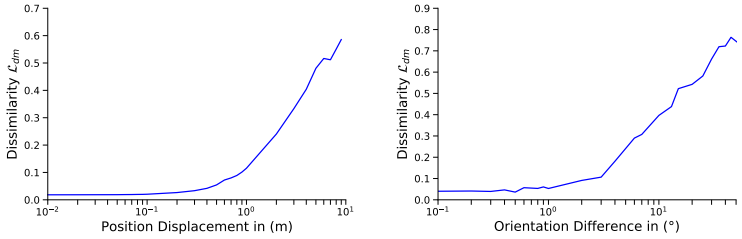


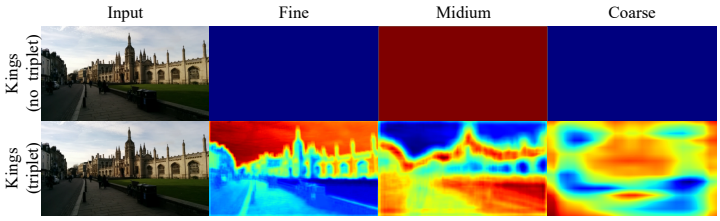
Fig. 5. Pose difference vs. feature dissimilarity. X-axis: camera position (**left**) and orientation difference (**right**) between a real image and a rendered image. Y-axis: feature dissimilarity $\mathcal{L}_{dm}$. Our direct feature matching loss $\mathcal{L}_{dm}$ is closely related to pose error, leading to effective training of the APR method.

4.4 Comparison to Sequential APR and 3D Approaches

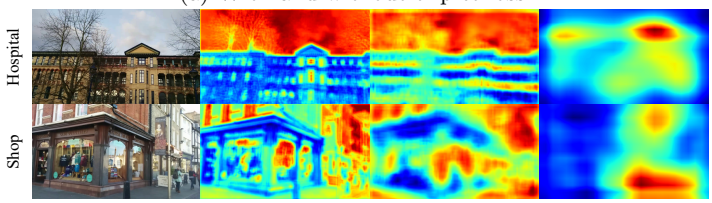
Table 3 compares our method to other types of relocalization approaches, such as several state-of-the-art sequential-based APR approaches and 3D structure-based method Active Search [27]. We notice that our DFNet$_{dm}$ outperforms most sequential-based APR methods except the translation error of VLocNet [34] on 7-Scenes in terms of the scene average performance. However, we still achieve superior accuracy than VLocNet in 7 out of 11 scenes. For the first time, the performance of single-image APR is comparable to 3D-structure methods. Our DFNet$_{dm}$ is slightly more accurate than Active Search [27] in average rotation error of 7-scenes. However, our method is still slightly behind in terms of translation error and Cambridge errors although by smaller margins.

4.5 Ablation Study

Effectiveness of Direct Feature Matching We run a toy example of direct feature matching on Shop Facade using finest features and combinations of multi-level features, as in Table 4(a). We discover that finer-level features are more



(a) With and without triplet loss



(b) More feature maps examples when trained with triplet loss

Fig. 6. **(a)** Top row: feature collapsing when training DFNet on Kings without using triplet loss. Bottom row: training DFNet with triplet loss can avoid the feature collapsing issue. **(b)** Feature maps of other scenes in Cambridge when training with triplet loss. We show that more refined level features consistently contain more meaningful details and, therefore more beneficial to use for direct feature matching.

Table 5. Data generation strategy comparison: RVS vs. LENS [22] on 7-Scenes. An EfficientNet backbone (as in LENS) is used in DFNet for a fair comparison. Our RVS strategy obtains a comparable results to LENS while using much less training data and rendering in much lower resolution, enabling online training.

Model	Backbone Top-1 Acc.	Pose Error (m/degree)	Real Data Quantity/Epoch	Synthetic Data Quantity/Epoch	Synthetic Resolution	Rendering Cost	Generation Mode
DFNet(VGG16)	71.59%	0.12/3.71	10-20%	10-20%	Low	Cheap	Online
DFNet(EB0)	76.3%	0.08/3.47	10-20%	10-20%	Low	Cheap	Online
LENS(EB3)	81.1%	0.08/3.00	71%-100%	710%-1000%	High	Expensive	Offline

helpful for direct matching. We believe this to be due to their capability to preserve high frequency details and sharper contents, as shown in (Fig. 6(b)). This explains why we only use the finest feature in the feature-metric direct matching implementation. Furthermore, Fig. 5 shows how the direct matching loss $\mathcal{L}_{dm}$ successfully correlates the pose differences to the feature similarity between real images and rendered images.

Features Collapse We demonstrate the difference when training DFNet's feature extractor with and without triplet loss in Fig. 6(a). We replace our triplet loss with a mean square error (MSE) loss for the without triplet loss case. Intuitively, losses that only minimize positive sample distances such as MSE, L_2, or L_1 losses may lead to feature collapsing [6] since the feature extraction blocks in DFNet are likely to learn to cheat. On the other hand, using triplet loss super-

vised with additional negative samples works well for extracting dense domain invariant features.

Summary of Ablation We break down our design decisions to show how each component contributes to the pose regression accuracy in Table 4(b). We start with training an DFNet model using with standard triplet loss without mining. The performance improves noticeably when we add the RVS. We also see around 16%/36% gain in translation and rotation errors when adding the customized triplet loss $\mathcal{L}_{triplet}$. We then validate our DFNet$_{dm}$'s direct feature matching (DFM), which further reduces error significantly. The DFM approach with histogram-assisted NeRF outperforms the NeRF-W one, which validates the effectiveness of our histogram embedding design. Finally, we attempt to train a Direct-PN+U model with our histogram-assisted NeRF modification. Our results show that the photometric direct matching-based method that can benefit from our new NVS method, though the pose estimation accuracy is worse than our feature-metric direct matching method.

Effectiveness of RVS Table 5 shows a comparison between our online RVS strategy with another peer work LENS [22] that uses NeRF data generation for APR training. Although both data generation methods effectively improve APR performance, our RVS strategy is a much cheaper alternative requiring lower rendering resolution (80x60 vs. 320x240 [22]) and fewer data. We are able to reach similar performance with LENS when we replace our VGG16 backbone with an EfficientNet-B0 [33], which proves that a simpler data generation strategy could also effectively improves APR methods.

5 Conclusion

In summary, we introduce an Absolute Pose Regression (APR) pipeline for camera re-localization. Specifically: 1) we propose a histogram-assisted NeRF to compensate dramatic exposure variance in large scale scene with challenging exposure conditions. The histogram-assisted NeRF, serving as a novel view renderer, enables a direct matching training scheme; 2) we explore a direct matching scheme in feature space, leading to a more robust performance than the photometric approach, and address a domain gap issue that arises when matching real images with synthetic images via a contrastive learning scheme; 3) we devise an efficient data generation strategy, which proposes pseudo training poses around existing training trajectories, leading to better generalization capability to unseen data. As a result, our method achieves a state-of-the-art accuracy by outperforming existing single-image APR methods by as much as 56%, comparable to 3D structure-based methods.

Acknowledgments The authors thank Michael Hobley, Theo Costain, Lixiong Chen, and Kejie Li for their thoughtful comments. Shuai Chen was supported by gift funding from Huawei.

References

1. Brachmann, E., Krull, A., Nowozin, S., Shotton, J., Michel, F., Gumhold, S., Rother, C.: DSAC - Differentiable RANSAC for Camera Localization. In: CVPR (2017)
2. Brachmann, E., Rother, C.: Learning Less is More - 6D Camera Localization via 3D Surface Regression. In: CVPR (2018)
3. Brachmann, E., Rother, C.: Visual camera re-localization from RGB and RGB-D images using DSAC. arXiv (2020)
4. Brahmbhatt, S., Gu, J., Kim, K., Hays, J., Kautz, J.: Geometry-Aware Learning of Maps for Camera Localization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
5. Chen, S., Wang, Z., Prisacariu, V.: Direct-PoseNet: Absolute pose regression with photometric consistency. In: 3DV (2021)
6. Chen, X., He, K.: Exploring simple siamese representation learning. In: CVPR (2021)
7. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: CVPRW (2018)
8. Engel, J., Koltun, V., Cremers, D.: DSO: Direct sparse odometry. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2017)
9. Engel, J., Schops, T., Cremers, D.: Semi-dense visual odometry for a monocular camera. In: In Proceedings of IEEE International Conference on Computer Vision (ICCV) (2013)
10. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: ICML (2017)
11. Glocker, B., Izadi, S., Shotton, J., Criminisi, A.: Real-time rgb-d camera relocalization. In: International Symposium on Mixed and Augmented Reality (ISMAR) (2013)
12. Kendall, A., Cipolla, R.: Modelling uncertainty in deep learning for camera relocalization. In: ICRA (2016)
13. Kendall, A., Cipolla, R.: Geometric loss functions for camera pose regression with deep learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
14. Kendall, A., Grimes, M., Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In: International Conference on Computer Vision (2015)
15. Kendall, A., Cipolla, R.: Modelling uncertainty in deep learning for camera relocalization. In: ICRA (2016)
16. Li, X., Han, K., Li, S., Prisacariu, V.: Dual-resolution correspondence networks. In: NeurIPS (2020)
17. Lindenberger, P., Sarlin, P.E., Larsson, V., Pollefeys, M.: Pixel-Perfect Structure-from-Motion with Featuremetric Refinement. ICCV (2021)
18. Martin-Brualla, R., Radwan, N., Sajjadi, M.S.M., Barron, J.T., Dosovitskiy, A., Duckworth, D.: NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In: CVPR (2021)
19. Melekhov, I., Ylioinas, J., Kannala, J., Rahtu, E.: Image-based localization using hourglass networks. In: ICCV Workshops (2017)
20. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)

21. Moreau, A., Piasco, N., Tsishkou, D., Stanciulescu, B., de La Fortelle, A.: Coordinet: uncertainty-aware pose regressor for reliable vehicle localization. In: arxiv preprint, arxiv:2103.10796 (2021)
22. Moreau, A., Piasco, N., Tsishkou, D., Stanciulescu, B., de La Fortelle, A.: LENS: Localization enhanced by nerf synthesis. In: CoRL (2021)
23. Ng, T., Lopez-Rodriguez, A., Balntas, V., Mikolajczyk, K.: Reassessing the limitations of cnn methods for camera pose regression. In: arXiv preprint arXiv:2108.07260 (2021)
24. Purkait, P., Zhao, C., Zach, C.: Synthetic view generation for absolute pose regression and image synthesis. In: BMVC (2018)
25. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: CVPR (2020)
26. Sarlin, P.E., Unagar, A., Larsson, M., Germain, H., Toft, C., Larsson, V., Pollefeys, M., Lepetit, V., Hammarstrand, L., Kahl, F., Sattler, T.: Back to the Feature: Learning robust camera localization from pixels to pose. In: CVPR (2021)
27. Sattler, T., Leibe, B., Kobbelt, L.: Improving image-based localization by active correspondence search. In: European conference on computer vision (2012)
28. Shavit, Y., Ferens, R., Keller, Y.: Learning multi-scene absolute pose regression with transformers. In: ICCV (2021)
29. Shavit, Y., Ferens, R., Keller, Y.: Paying attention to activation maps in camera pose regression. In: arXiv preprint arXiv:2103.11477 (2021)
30. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in rgb-d images. In: CVPR (2013)
31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
32. Taira, H., Okutomi, M., Sattler, T., Cimpoi, M., Pollefeys, M., Sivic, J., Pajdla, T., , Torii, A.: InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. CVPR (2018)
33. Tan, M., EfficientNet, Q.L.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: PMLR (2019)
34. Valada, A., Radwan, N., Burgard, W.: Deep auxiliary learning for visual localization and odometry. In: ICRA (2018)
35. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS (2017)
36. Walch, F., Hazirbas, C., Leal-Taixe, L., Sattler, T., Hilsenbeck, S., Cremers, D.: Image-based localization using lstms for structured feature correlation. In: International Conference on Computer Vision (2017)
37. Wang, Z., Wu, S., Xie, W., Chen, M., Prisacariu, V.A.: NeRF−−: Neural radiance fields without known camera parameters. arXiv preprint arXiv:2102.07064 (2021)
38. Wu, J., Ma, L., Hu, X.: Delving Deeper into Convolutional Neural Networks for Camera Relocalization. In: ICRA (2017)
39. Yen-Chen, L., Florence, P., Barron, J.T., Rodriguez, A., Isola, P., Lin, T.Y.: inerf: Inverting neural radiance fields for pose estimation. In: arxiv arXiv:2012.05877 (2020)

6 Supplementary

6.1 Implementation

Histogram-assisted NeRF Here, we provide more implementation details of our methods. As we mentioned in Section 3.3 of the main paper, our histogram-assisted NeRF model renders at a speed of 12.2 fps (benchmarked by a 3080Ti GPU) to achieve online RVS training. In order to achieve a balanced trade-off between speed and quality, we choose to render small images with a shorter side of 60 pixels. In addition, we set the NeRF model architecture to 64 coarse and 64 fine sampling with an MLP width of 128.

DFNet Our DFNet takes an image input with a shorter side of 240 pixels. For feature extractor module $\mathcal{G}$ of DFNet, features are fed through a Conv-Relu-Conv-Batch Norm architecture with 64 kernels and 128 output channels. The DFNet is trained with a batch size of 4 or 8, depending on the GPU's memory. We implement an early stopping strategy with a patient value of 200 and schedule the learning rate decay of 0.95 when validation loss plateaus for every 50 epochs. For every $N = 20$ epochs, we will randomly generate the same amount of views as the training sample size using RVS.

Direct Feature Matching To train the DFNet$_{dm}$ model with feature-metric direct matching using unlabeled data, we set the batch size to 1 and the learning rate to 1×10^{-5} with the same early stopping strategy mentioned above. We discover that only low-level features (i.e., features from the first blocks of VGG) are needed to achieve the best performance, which we discussed earlier in Section 4.5 of the main paper.

6.2 Visualization

Qualitative Comparison on 7-Scenes We show a selection of qualitative comparisons on the 7-Scenes dataset with several baseline APR methods [13,4,5] in Fig. 7. We also encourage our readers to check out our supplementary video, in which we rendered views of the predicted pose using NeRF synthesis.

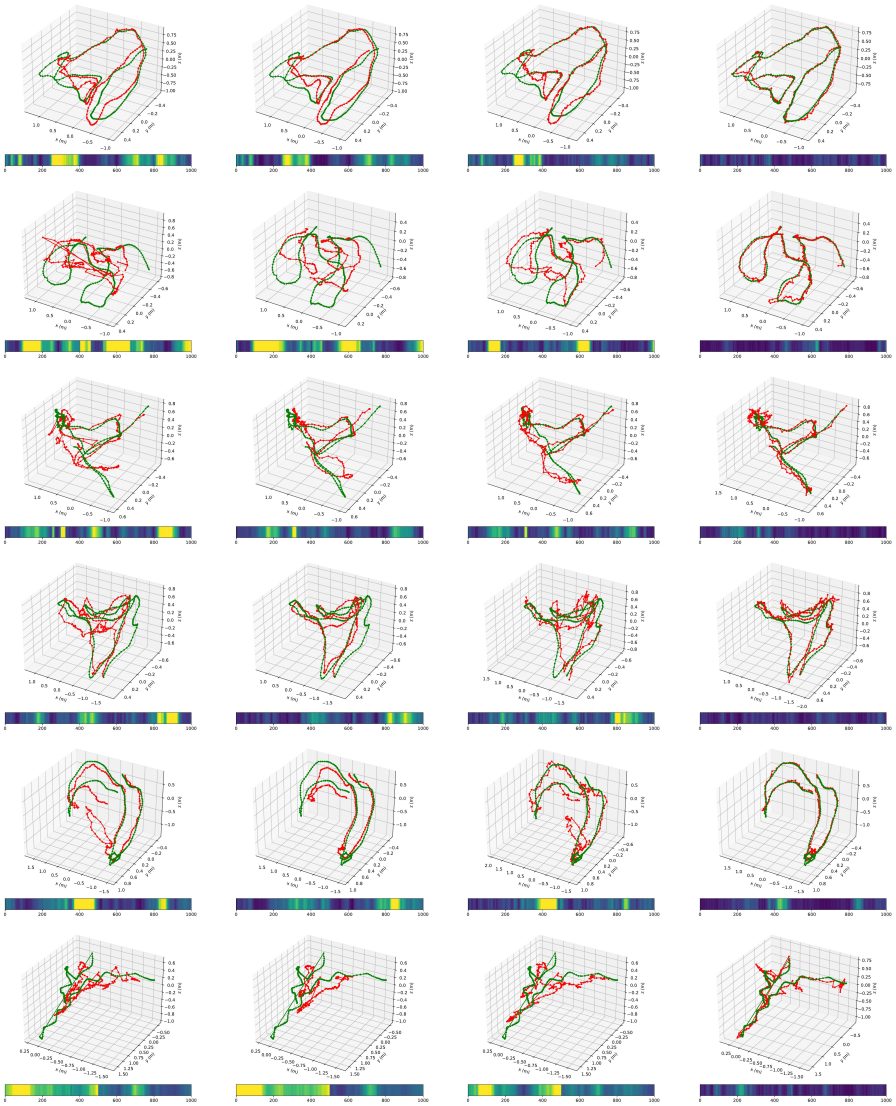
NeRF-W vs. Histogram-assisted NeRF In real-life camera localization applications, since training and testing data are likely to be taken from different sequences, camera exposures, or time of the day, our histogram-assisted NeRF would be more helpful to render accurate appearcance Fig. 8. We experimentally found our histogram-assisted NeRF is helpful in both photometric matching and feature-metric matching approaches.

6.3 Additional Discussion

Photometric Distortion As discussed in section 3.2 of the main paper, photometric matching relies on RGB-wise differences between the query and rendered images. However, if those images appear in different lighting/exposure conditions, the photometric loss will fail due to large RGB-wise differences even under the same camera pose. Previous photometric matching approaches, such as Direct-PN+U, perform worse in pose estimation when using unlabeled data with large appearance variations from the training sequences (refer to the lower part of main paper Table 4b). We observed that such degradation is consistent in other outdoor scenes.

Two Properties of Domain Invariant Features We had two clear goals for designing our robust feature extractor: (1) We want the extracted features to be sensitive to pose changes. (2) We want the features to be indistinguishable between real and rendered image features from the same pose (Close the Domain Gap). Our first goal is achieved by the L_2 pose loss supervision, which ensures the features are closely related to the pose regression task. We specifically design the Feature Extractor to share the backbone with the Pose Module (see main paper Fig 2a). Although the deeper layer features may lose semantic meaning, we observe that those features can respond to pose changes.

The triplet loss is primarily designed to achieve the second goal without feature collapse in the training process. We previously tried to force real and rendered image features to be the same by using MSE/L_2 losses, leading to feature collapse (main paper Fig 6a). This is because the pink layers in main paper Fig 2a, despite being shallow, are likely to learn to cheat since those layers are not supervised by other meaningful losses. Thus, we introduce the triplet loss to prevent features collapsing. We experimentally find that the proposed in-triplet mining adds extra robustness to both feature extraction and pose regression and leads to better APR performance overall. Such observation could hint that removing the domain gap benefits APR training when using extra randomly generated synthetic training data.



(a) PoseNet[14,12,13] (b) MapNet+PGO[4] (c) Direct-PN+U[5] (d) DFNet$_{dm}$

Fig. 7. Qualitative comparison on the 7-Scenes dataset. The 3D plots show the camera positions, green for ground truth and red for predictions. The bottom color bar represents rotational errors for each subplot, where yellow means large error and blue means small error for each test sequence. Sequence names from top to bottom are: Chess-seq-03, Fire-seq-04, Office-seq-07, Kitchen-seq-06, Kitchen-seq-12, Stairs-all.

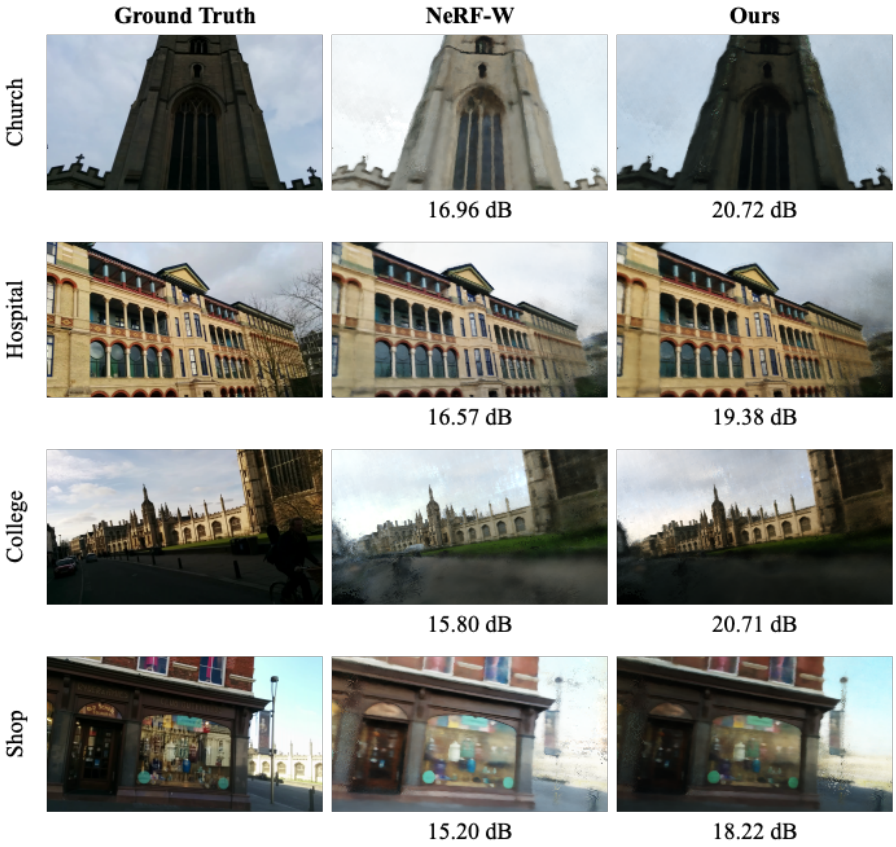


Fig. 8. A visual comparison between NeRF-W and our histogram-assisted NeRF on the testing sequences of Cambridge Landmarks dataset. The corresponding scene's test PSNR is displayed at the bottom of each sub-figure.

Statement of Authorship for the paper "DFNet: Enhance Absolute Pose Regression with Direct Feature Matching" in Chapter 4.

Paper title	DFNet: Enhance Absolute Pose Regression with Direct Feature Matching
Authors	**Shuai Chen**, Xinghui Li, Zirui Wang, Victor Adrian Prisacariu
Publication status	Published
Publication details	European Conference on Computer Vision (ECCV), 2022.

Student Confirmation

Student name	Shuai Chen	
Contribution to the paper	First-author contribution: • Conception of research ideas • Design and implementation of models • Running of large-scale experiments • Writing and presentation of the paper	
Signature and Date		Apr. 23th 2025

Supervisor Confirmation

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

Supervisor name	Prof. Victor Adrian Prisacariu	
Supervisor comments	The description is accurate	
Signature and Date		Apr. 23th 2025

Part II

Inference of Absolute Pose Regression

Chapter 5

Neural Pose Refinement via Neural Feature Fields

This chapter explores an inference-based camera pose refinement method for visual relocalisation, leveraging the scene information embedded within the Neural Feature Fields as a geometric prior. To be more precise, this can be be further categorised as an implicit prior. Unlike explicit 3D models (*e.g.*, point clouds or meshes), this prior is implicitly encoded within the neural network's parameters, capturing the underlying geometric and semantic consistency of the scene to guide the pose refinement process.

While APR methods can obtain improved accuracy by incorporating 3D geometric supervision during training, as shown in Chapter 3 and Chapter 4, they are unable to leverage 3D geometric information at test time. This lack of test-time adaptability limits their robustness and contributes to their suboptimal performance compared to state-of-the-art 3D geometry-based approaches.

Motivated by this challenge, this chapter introduces Neural Feature Synthesizer (NeFeS), a test-time refinement framework designed to enhance the prediction of any APR method. Specifically, NeFeS is APR-agnostic and operates without

requiring additional labelled or unlabelled data, making it compatible with a wide range of existing APR architectures.

NeFeS consists of several key elements. First, it learns a volumetric 3D feature field that encodes geometric structure through differentiable rendering during training. At inference time, NeFeS directly renders dense novel view features from novel viewpoints corresponding to initial APR predictions to refine the poses. Second, NeFeS integrates a Feature Fusion Module and an exposure compensation method called Affine Color Transformation (ACT) to improve the quality of the rendered features. A progressive training strategy is used to further stabilise the feature learning process.

We conducted extensive experiments to show that NeFeS consistently improves the accuracy of various APR baselines and achieves state-of-the-art performance across indoor and outdoor benchmarks. Notably, it does so without requiring any test-time supervision or retraining of the existing APR model.

The main content presented below in this chapter is based on our paper, *Neural Refinement for Absolute Pose Regression with Feature Synthesis*, published at the *2024 Conference on Computer Vision and Pattern Recognition*.

Neural Refinement for Absolute Pose Regression with Feature Synthesis

Shuai Chen[1] Yash Bhalgat[2] Xinghui Li[1] Jia-Wang Bian[1]
Kejie Li[1] Zirui Wang[1] Victor Adrian Prisacariu[1]
[1]Active Vision Lab, University of Oxford
[2]Visual Geometry Group, University of Oxford

Abstract

Absolute Pose Regression (APR) methods use deep neural networks to directly regress camera poses from RGB images. However, the predominant APR architectures only rely on 2D operations during inference, resulting in limited accuracy of pose estimation due to the lack of 3D geometry constraints or priors. In this work, we propose a test-time refinement pipeline that leverages implicit geometric constraints using a robust feature field to enhance the ability of APR methods to use 3D information during inference. We also introduce a novel Neural Feature Synthesizer (NeFeS) model, which encodes 3D geometric features during training and directly renders dense novel view features at test time to refine APR methods. To enhance the robustness of our model, we introduce a feature fusion module and a progressive training strategy. Our proposed method achieves state-of-the-art single-image APR accuracy on indoor and outdoor datasets. Code will be released at https://github.com/ActiveVisionLab/NeFeS.

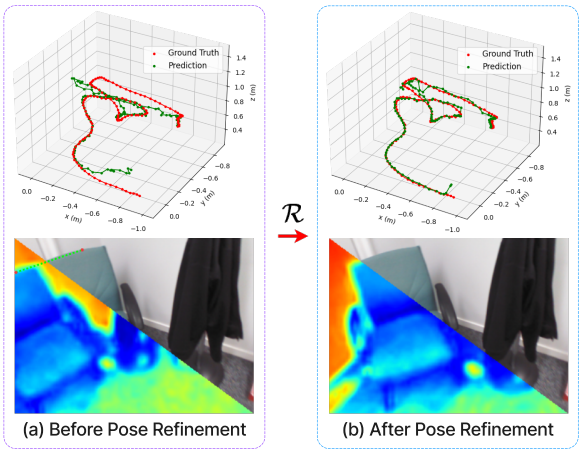


Figure 1. Our pose refinement ($\mathcal{R}$) improves (coarse) pose predictions from other methods using novel feature synthesis to achieve pixel-wise alignment. **Top left / right:** 3D plots of predicted (**green**) and ground-truth (**red**) camera positions. **Bottom left / right:** alignment between rendered features and query image.

1. Introduction

Camera relocalization is a crucial task that allows machines to understand their position and orientation in 3D space. It is an essential prerequisite for applications such as augmented reality, robotics, and autonomous driving, where the accuracy and efficiency of pose estimation are important. Recently, Absolute Pose Regression (APR) methods [21–23] have been shown to be effective in directly estimating camera pose from RGB images using convolutional neural networks. The simplicity of APR's architecture offers several potential advantages over classical geometry-based methods [5, 46, 48], involving end-to-end training, cheap computation cost, and low memory demand.

Latest advances in APR, particularly the use of novel view synthesis (NVS) [10, 11, 29, 32, 33, 52] to generate new images from random viewpoints as data augmentation during training, have significantly improved the pose regression performance. Despite this, state-of-the-art (SOTA) APRs still have the following limitations: (i) They predict the pose of a query image by passing it through a CNN, which typically disregards geometry at inference time. This causes APR networks to struggle to generalize to viewpoints that the training data fails to cover [49]; (ii) The unlabeled data, often sampled from the validation/testing set, used for finetuning the APR network [8, 10, 11] may not be universally available in real-life circumstances, and this semi-supervised finetuning is also time-consuming.

To address these limitations, we propose a novel test-time refinement pipeline for APR methods. Unlike prior works that explore extended Kalman filters [34], pose graph optimization [8], or pose auto-encoders [52], our method integrates an *implicit* representation based geometric refinement into an end-to-end learning framework, where gradients can be backpropagated to the APR network. We test our proposed method across different APR architectures to

demonstrate its robustness and effectiveness. Furthermore, we propose a Neural Feature Synthesizer (NeFeS) network to encode the 3D geometry of a scene implicitly into an MLP. NeFeS renders dense features from novel viewpoints for refinement. To ensure the robustness of feature rendering, we introduce a Feature Fusion module into NeFeS that combines the rendered color and features and is trained in a progressive manner. Our method leverages prior literature on volume rendering to inherently constrain geometric consistency during test time using implicit 3D neural feature fields. As such, our approach occupies a middle ground between APR and methods informed by geometry.

We summarize our main contributions as follows: **First**, we propose a test-time refinement pipeline that greatly improves the pose-estimation accuracy of any APR model without using additional unlabeled data and exhibits a new *single-frame* APR SOTA performance on standard benchmarks. **Second**, we propose a Neural Feature Synthesizer (NeFeS) network that implicitly encodes 3D geometric features. NeFeS refines an initial pose by rendering a dense feature map and making the comparison with the query image feature. **Third**, we propose a progressive training strategy and a Feature Fusion module to improve the robustness of the rendering ability of the NeFeS model.

2. Related Work

Absolute Pose Regression (APR). APR methods have been widely studied due to their simple and lightweight formulation that allows the camera pose to be directly regressed using an end-to-end neural network. PoseNet [21–23] introduced the first APR solution using GoogLeNet-backbone, followed by various architectures like the hourglass network [31], attention layers [53, 54, 64], separated translation and rotation prediction [37, 66], or LSTM [63].

To further improve APR accuracy, some works utilize sequential information. These approches incorporate temporal constraints such as visual odometry [8, 42, 61], motion [34], temporal filtering [15], and multitasking [42]. Recent APR methods also benefit from novel view synthesis, where one line of approaches focuses on generating large amounts of extra photo-realistic synthetic data [11, 33, 40] via randomly sampled virtual camera poses. However, generating high-quality offline synthetic data may take several days [33] for each scene. Other approaches [10, 11] use NeRF [29, 32] as a direct matching module to perform unlabeled finetuning [8] using extra images without ground-truth pose annotation. However, finetuning takes significant time and assumes that extra unlabeled data can be easily obtained.

While the aforementioned works enhance APR training, we focus on improving generic APR methods during test time. Unlike prior works that only exam means for test-time refinement on a single specific APR architecture, such as extended Kalman filters [34], pose graph optimization [8], or pose auto-encoders [52], our method exhibits strong flexibility to be adapted to a wide range of APR architectures on both camera positions and orientations, achieving state-of-the-art results without extra unlabeled data.

Notably, classical geometry-based techniques [4–6, 28, 44, 46–48] that require explicit feature correspondence search [16, 17, 26, 45, 57, 58] also employ test-time refinement to improve localization accuracy. For example, [28, 44, 46] build upon image retrievals and pre-computed SfM model to perform standard geometric refinement via neural network-based feature matcher, PnP+RANSAC, or dense featuremetric-alignment. Our method, however, offers end-to-end neural feature refinement via implicit representation, enhancing existing APR models without external storage, pre-computed data, or manual tuning.

Neural Radiance and Feature Fields. Neural Radiance Fields (NeRF) [32] revolutionized novel view image synthesis and 3D surface reconstruction. NeRF's implicit 3D representation and differentiable volume rendering enable self-supervised optimization from RGB images, avoiding costly 3D annotations. iNeRF [67] showed that NeRF can be inverted for pose optimization. Recent approaches such as BARF [27] and its counterparts [3, 14, 65] simultaneously train NeRF by treating camera poses as learnable parameters in simple, non-360°scenes. Parallel works, NICE-SLAM [69] and iMAP [56], use NeRF for dense geometry and real-time camera tracking. Direct-PN [10] uses NeRF as a direct matching module to compute the photometric errors and propagate the error gradients back to the pose regression network. DFNet [11] extends this method to outdoor scenarios with robust feature extraction. LENS [33] uses NeRF to generate a synthetic training dataset based on manually tuned scene bounds and parameters.

Recently, NeRF models have been extended to directly predict and render *feature fields* alongside density and appearance fields. Typically, these feature fields are learned by supervision from a 2D feature extractor using volumetric rendering. [2, 24, 60] showed that these 3D feature fields outperform 2D baselines [9, 12, 25] on downstream tasks such as 2D object retrieval or 3D segmentation. CLIP-Fields [51] established feature fields as scene memory for robot navigation. This work explores distilled neural feature fields for camera relocalization, highlighting their role in test-time pose refinement.

3. Method

In this section, we present a detailed outline of our approach. Sec. 3.1 provides a high-level overview of our refinement framework. Sec. 3.2 describes the architecture and training details of our proposed NeFeS network along with its two components: *Exposure-adaptive Affine Color Transformation (ACT)* and *Feature Fusion module*.

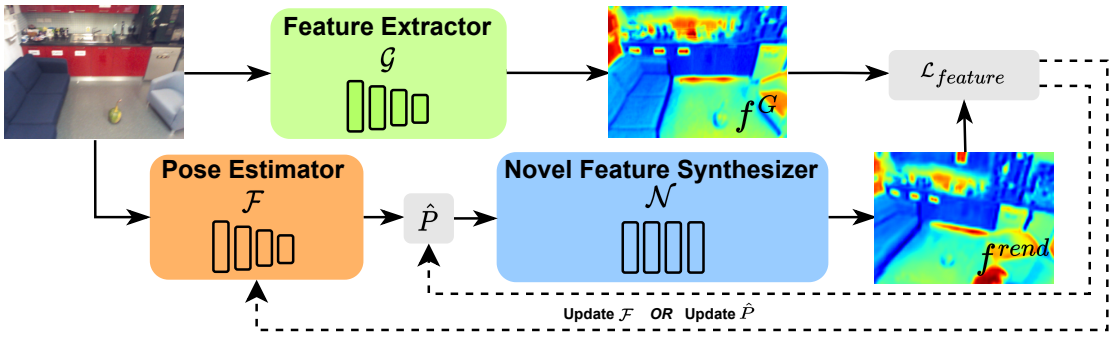


Figure 2. **Illustration of the pose refinement pipeline.** The query image is processed by a pose estimator $\mathcal{F}$, typically an absolute pose regressor, to obtain a coarse camera pose $\hat{P}$. Our novel feature synthesizer $\mathcal{N}$ renders a dense feature map f^{rend} based on $\hat{P}$. Simultaneously, the feature extractor $\mathcal{G}$ extracts the feature map f^G from the query image. We then compute the feature-metric error between f^{rend} and f^G, denoted as $\mathcal{L}_{feature}$. This error is backpropagated to update either the parameters of $\mathcal{F}$ or the pose $\hat{P}$ directly.

3.1. Refinement Framework for APR

Given a query image I, an absolute pose regression (APR) network $\mathcal{F}$ directly regresses the camera pose $\hat{P}$ of I: $\hat{P} = \mathcal{F}(I)$. The network is typically trained with ground truth image-pose pairs. While APR-based methods are much more efficient than geometry-based methods since they require only a single forward pass of the network, the quality of their predictions is often significantly worse than those of geometry-based methods due to the lack of any 3D geometry-based reasoning [49].

In contrast to prior APR research, which attempts to improve APR by adding constraints to the training loss or making architectural changes to the backbone network, we propose an alternative method to refine the results of APR methods by backpropagating a feature-metric error at inference time. Our method has three major components (see Fig. 2): (1) a pretrained APR network, denoted as $\mathcal{F}$, which provides an initial pose; (2) a differentiable novel feature synthesizer $\mathcal{N}$ that directly renders dense feature maps given a camera pose; (3) an off-the-shelf feature extractor $\mathcal{G}$ that extracts the dense feature map of the query image. In our implementation, the feature extraction module from [11] is employed as the feature extractor $\mathcal{G}$.

The refinement procedure is as follows: (i) The query image I is passed through the pretrained APR model $\mathcal{F}$ to predict a coarse camera pose $\hat{P}$. (ii) The feature synthesizer $\mathcal{N}$ renders a dense feature map $f^{rend} \in \mathbb{R}^{n \times c}$ given the coarse camera pose $\hat{P}$, where $n = h \times w$, and h and w are the spatial dimensions of the feature map[1]. (iii) At the same time, the feature extractor $\mathcal{G}$ extracts a feature map $f^G = \mathcal{G}(I)$ from the query image, where $f^G \in \mathbb{R}^{n \times c}$. (iv) The pose $\hat{P}$ is iteratively refined by minimizing the feature cosine similarity loss $\mathcal{L}_{feature}$ [11] between f^{rend} and f^G:

$$\mathcal{L}_{feature} = \sum_{i=1}^{c} \left(1 - \frac{\langle f^{rend}_{:,i} \cdot f^{G}_{:,i} \rangle}{\|f^{rend}_{:,i}\|_2 \cdot \|f^{G}_{:,i}\|_2} \right) \quad (1)$$

where $f^{rend}_{:,i}, f^{G}_{:,i} \in \mathbb{R}^n$, $\langle \cdot, \cdot \rangle$ denotes the inner product between two vectors and $\| \cdot \|_2$ represents the L2 norm. Different from the common feature matching literature's [26, 57] convention, our features are normalized along the spatial direction instead of the channel direction to ensure the consistency of the neighboring pixels.

Our method can be regarded as post-processing to the initial pose $\hat{P}$. We do not save the updated weights of the APR method since we restart from the initial state when given a new query image.

3.2. Neural Feature Synthesizer

We propose a Neural Feature Synthesizer (NeFeS) model that directly renders dense feature maps of a given viewpoint to refine the predictions of an underlying APR network. Similar to NeRF-W [29], our NeFeS architecture uses a base MLP module with *static* and *transient* heads that predict the static and transient density ($\sigma^{(s)}$ and $\sigma^{(\tau)}$) and view-dependent color ($c^{(s)}$ and $c^{(\tau)}$) respectively, given an input 3D position ($\mathbf{x}$) and viewing direction ($\mathbf{d}$). We use the frequency encoding [32, 62] to encode all 3D positions and view directions. The transient head models the colors of the 3D points using an isotropic normal distribution and predicts a view-dependent variance value (β^2) for the transient color distribution. To render the color of a given pixel, the original volume rendering formulation in NeRF [32] is augmented to include the transient colors and densities, and the color of a given image-pixel ($\hat{\mathbf{C}}(\mathbf{r})$) is computed as a composite of the static and transient components. Here, $\mathbf{r}$ denotes the ray (corresponding to the pixel) on which points are sampled to compute the volume rendering quadrature approximation [30]. The variances of sampled points along

[1]Note: We treat the n dimension as the feature rather than c dimension.

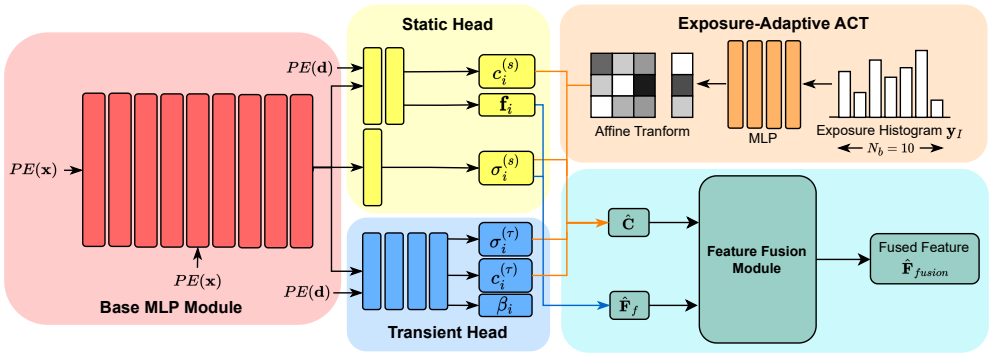


Figure 3. The architecture of our proposed NeFeS model. The query 3D position $\mathbf{x}$ is fed to the network after positional encoding $PE(\cdot)$. The network then splits into two heads: the static head and the transient head. Given a viewing direction $\mathbf{d}$, the rendered color map is generated by fusing static RGB value $c_i^{(s)}$, the transient RGB value c_i^{τ} and their corresponding density values $\sigma_i^{(s)}$ and σ_i^{τ}, while the rendered feature map is formed only by static features $\mathbf{f}_i$ and density $\sigma_i^{(s)}$. In addition, the color map adopts exposure-adaptive ACT to compensate for exposure differences between images. The final feature map $\hat{\mathbf{F}}_{fusion}$ is the concatenation of rendered RGB and feature map processed by the feature fusion module.

the corresponding ray are also rendered using only the transient densities (and *not* the static densities) to obtain a per-pixel color variance $\beta(\mathbf{r})^2$. We refer reader to [29] for more details on the static+transient volume rendering.

We expand the output of the static MLP to also predict features for an input 3D position. The output dimension is $N_c + N_f$, where N_f features are predicted along with RGB values. The per-pixel features are rendered using the same volume rendering quadrature approximation [30]:

$$\hat{\mathbf{F}}_f(\mathbf{r}) = \sum_{i=1}^{N} T_i \left(1 - \exp\left(-\sigma_i^{(s)}\delta_i\right)\right)\mathbf{f}_i,$$
$$T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j^{(s)}\delta_j\right) \tag{2}$$

where $\mathbf{f}_i$ and $\sigma_i^{(s)}$ are the feature and density predicted by the static MLP for a sampled point on the ray, and δ_i is the distance between sampled quadrature points i and $i+1$.

Fig. 3 demonstrates the architecture of our proposed NeFeS model. We propose two crucial components in the rendering pipeline of our NeFeS architecture that ensure the robustness of our rendered features.

Exposure-adaptive ACT. In the context of camera relocalization, testing images may differ in exposure or lighting from training sequences. To address this, DFNet [11] proposed using the luminance histogram of the query image as a latent code input to the color prediction head of the NeRF MLP. However, since our NeFeS outputs both colors and features simultaneously, we find this approach perturbs the feature output values and causes instability. Ideally, the feature descriptors should be able to maintain local invariance even under varying exposure. Inspired by Urban Radiance Fields (URF) [43], we propose to use an *exposure-adaptive Affine Color Transformation (ACT)* which is a 3×3 matrix $\mathbf{K}$ and a 3-dimensional bias vector $\mathbf{b}$ predicted by a 4-layer MLP with the query image's luminance histogram $\mathbf{y}_I$. Unlike URF, which uses a pre-determined exposure code, we use the query image's histogram embedding for accurate appearance rendering of unseen testing images. The final per-pixel color $\hat{\mathbf{C}}(\mathbf{r})$ is computed using the affine transformation as $\hat{\mathbf{C}}(\mathbf{r}) = \mathbf{K}\hat{\mathbf{C}}_{rend}(\mathbf{r}) + \mathbf{b}$, where $\hat{\mathbf{C}}_{rend}(\mathbf{r})$ is the rendered per-pixel color obtained using the static and transient MLPs.

Feature Fusion Module We propose a Feature Fusion module to fuse the rendered colors and features to produce the final feature map. The rendered colors and features are concatenated and fed into the fusion module consisting of three 3x3 convolutions, followed by a 5x5 convolution and a batch normalization layer. During inference, we render colors and features for all $H \times W$ image pixels and the resulting $H \times W \times (N_c + N_f)$ tensor is processed by the module. Note, for efficiency during training, we sample $S \times S$ regions to render and apply the loss to those pixels each iteration.

We use $\mathcal{H}$ to represent the fusion module. The final output feature result is:

$$\hat{\mathbf{F}}_{fusion}(\mathcal{R}) = \mathcal{H}(\hat{\mathbf{C}}(\mathcal{R}), \hat{\mathbf{F}}_f(\mathcal{R})) \tag{3}$$

where $\mathcal{R}$ is the sampled region as described above.

We experimentally find that the fusion module produces more robust features than the input rendered features $\hat{\mathbf{F}}_f$. We refer readers to the supplementary for detailed ablations.

Training the Feature Synthesizer The high-level concept of training the NeFeS is motivated by feature field distillation proposed in [24], which essentially distills the 2D backbone features into a 3D NeRF model. However, 2D features in our NeFeS need to be closely related to the direct matching formulation [10, 20]. In this work, we use

the trained 2D feature extractor from [11] to produce the feature labels due to its effectiveness in generating domain invariant features.

Loss Functions. The total loss used to train our NeFeS model consists of a photometric loss $\mathcal{L}_{rgb}$ and two l_1-based feature-metric losses:

$$\mathcal{L} = \mathcal{L}_{rgb} + \lambda_1 \mathcal{L}_f + \lambda_2 \mathcal{L}_{fusion}. \tag{4}$$

The photometric loss is defined as the negative log-likelihood of a normal distribution with variance $\beta(\mathbf{r})^2$:

$$\begin{aligned} \mathcal{L}_{rgb}(\mathbf{r}) = & \frac{1}{2\beta_i(\mathbf{r})^2} \left\| \mathbf{C}(\mathbf{r}) - \hat{\mathbf{C}}(\mathbf{r}) \right\|_2^2 \\ & + \frac{1}{2} \log \beta(\mathbf{r})^2 + \frac{\lambda_s}{K} \sum_{k=1}^{K} \sigma_k^{(\tau)} \end{aligned} \tag{5}$$

where $\mathbf{r}$ is the ray direction corresponding to an image pixel, $\mathbf{C}_i(\mathbf{r})$ and $\hat{\mathbf{C}}_i(\mathbf{r})$ are the ground-truth and rendered pixel colors. The third term in Eq. (5) is a sum of the transient densities of all the points on ray $\mathbf{r}$ and is used to ensure that transient densities are sparse.

The feature losses are simply l_1 losses:

$$\mathcal{L}_f = \sum_{\mathbf{r} \in \mathcal{R}} \| \hat{\mathbf{F}}_f(\mathbf{r}) - \mathbf{F}_{\mathbf{img}}(I, \mathbf{r}) \|_1, \tag{6}$$

and

$$\mathcal{L}_{fusion} = \sum_{\mathbf{r} \in \mathcal{R}} \| \hat{\mathbf{F}}_{fusion}(\mathbf{r}) - \mathbf{F}_{\mathbf{img}}(I, \mathbf{r}) \|_1. \tag{7}$$

where $\mathbf{F}_{\mathbf{img}}(I, \cdot)$ are the features extracted from the training images using the pre-trained 2D feature extractor [11]. Note that, $\mathcal{L}_f$ is applied to the rendered features $\hat{\mathbf{F}}_f$ and $\mathcal{L}_{fusion}$ is applied to the fused features $\hat{\mathbf{F}}_{fusion}$. We experimentally find that using l_1 gives more robust features than l_2 and cosine feature loss for the test time refinement.

Progressive Training. We propose using a progressive schedule to train the NeFeS model. We first train the color and density part of the network for T_1 epochs to bootstrap the correct 3D geometry for the network. For these epochs, only $\mathcal{L}_{rgb}$ is used. Then we add $\mathcal{L}_f$ with weight λ_1 for the next T_2 epochs to train the feature part of the static MLP. Since the ground-truth features may not be fully multi-view consistent, we apply *stop-gradients* to the predicted density for the feature rendering branch. And finally, we add the feature fusion loss $\mathcal{L}_{fusion}$ with weight λ_2 for the last T_3 epochs. Since the feature fusion module takes both RGB images and 2D features as input, we randomly sample N_{crop} patches of $S \times S$ regions of the image and features to increase training efficiency. According to our experiments, this progressive training schedule leads to better convergence and performance. In addition, we apply semantic filtering to improve the network training results. Specifically, we use an off-the-shelf panoptic segmentation method [13] to mask out temporal objects in the scene such as people and moving vehicles.

3.3. Direct Pose Refinement

While our method is primarily designed to optimize APR, it is also possible to directly optimize camera pose parameters. We explore this feature by showing a possible scenario wherein the source of the pose estimation is either a black box or cannot be optimized (e.g. the initial camera pose comes from image retrieval). In these settings, we can set up our proposed method to directly refine the camera poses. Specifically, given an estimated camera pose $\hat{P} = [\mathbf{R}|\mathbf{t}]$, where $\mathbf{R}$ is rotation and $\mathbf{t}$ is the translation component, our method optimizes the camera poses using tangent space backpropagation[2]. Additionally, we found that using two different learning rates for the translation and rotation parts helps achieve faster and more stable convergence for camera pose refinement. This is different from the standard convention used in [3, 14, 27, 65] . We refer our readers to supplementary material for more details.

4. Experiments

We implement our method in PyTorch [39]. Implementation details about the NeFeS architecture and progressive training scheduling can be found in the supplementary[3].

4.1. Evaluation on Cambridge Landmarks

We evaluate our proposed refinement method on Cambridge Landmarks [23] , which is a popular outdoor dataset used for benchmarking pose regression methods. The dataset contains handheld smartphone images of scenes with large exposure variations and covers an area of $875m^2$ to $5600m^2$. The training sequences contain 200-1500 samples, and test sets are captured from different sequences. For each test image, we refine the model using the approach in Sec. 3.1 for $m = 50$ iterations.

We first test our method on top of an open-sourced SOTA single-frame APR method. Tab. 1 summarizes the results of our method and existing APR methods. Our method achieves the best accuracy across all four scenes when coupled with the DFNet. In Sec. 4.3, we demonstrate the performance of our method with other APR approaches. Particularly, our method improves DFNet by as much as 70.6% compared to its scene average results. All the per-scene performances from the other compared methods are taken from their papers, except for MS-Transformer+PAE, which only reports the scene average median errors. We encourage our readers to check out supplementary for more thorough comparisons and ablations to the Cambridge Landmarks dataset.

[2] We use the LieTorch [59] library for this.

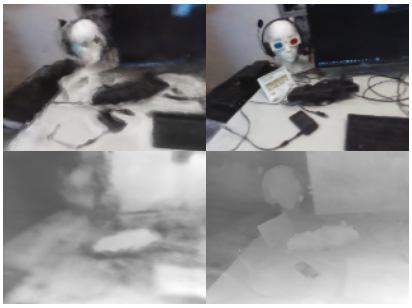
[3] Supplementary: Implementation Details

Methods	Kings	Hospital	Shop	Church	Average
PoseNet(PN)[23]	1.66/4.86	2.62/4.90	1.41/7.18	2.45/7.96	2.04/6.23
PN Learn σ^2[22]	0.99/1.06	2.17/2.94	1.05/3.97	1.49/3.43	1.43/2.85
geo. PN[22]	0.88/1.04	3.20/3.29	0.88/3.78	1.57/3.32	1.63/2.86
LSTM PN[63]	0.99/3.65	1.51/4.29	1.18/7.44	1.52/6.68	1.30/5.51
MapNet[8]	1.07/1.89	1.94/3.91	1.49/4.22	2.00/4.53	1.63/3.64
TransPoseNet[53]	0.60/2.43	1.45/3.08	0.55/3.49	1.09/4.94	0.91/3.50
MS-Transformer[54]	0.83/1.47	1.81/2.39	0.86/3.07	1.62/3.99	1.28/2.73
MS-Transformer+PAE[52]	-	-	-	-	0.96/2.73
DFNet [11]	0.73/2.37	2.00/2.98	0.67/2.21	1.37/4.03	1.19/2.90
DFNet + **NeFeS$_{50}$(ours)**	**0.37/0.54**	**0.52/0.88**	**0.15/0.53**	**0.37/1.14**	**0.35/0.77**

Table 1. **Comparisons on Cambridge Landmarks.** We compare our proposed test-time refinement method with single-frame APR methods. The subscript of **NeFeS$_{50}$** denotes the number of optimization iterations used for APR refinement. We report the median position and orientation errors in $m/°$. The best results are highlighted in **bold**.

Methods	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Average
PoseNet	0.10/4.02	0.27/10.0	0.18/13.0	0.17/5.97	0.19/4.67	0.22/5.91	0.35/10.5	0.21/7.74
MapNet	0.13/4.97	0.33/9.97	0.19/16.7	0.25/9.08	0.28/7.83	0.32/9.62	0.43/11.8	0.28/10.0
MS-Transformer	0.11/6.38	0.23/11.5	0.13/13.0	0.18/8.14	0.17/8.42	0.16/8.92	0.29/10.3	0.18/9.51
PAE	0.13/6.61	0.24/12.0	0.14/13.0	0.19/8.58	0.17/7.28	0.18/8.89	0.30/10.3	0.19/9.52
DFNet	0.03/1.12	0.06/2.30	0.04/2.29	0.06/1.54	0.07/1.92	0.07/1.74	0.12/2.63	0.06/1.93
DFNet + **NeFeS$_{50}$(ours)**	**0.02/0.57**	**0.02/0.74**	**0.02/1.28**	**0.02/0.56**	**0.02/0.55**	**0.02/0.57**	**0.05/1.28**	**0.02/0.79**

Table 2. **Comparisons on 7-Scenes dataset.** We compare the proposed refinement method with previous single-frame APR methods. We evaluate all methods with SfM ground truth poses provided in [7], measured in median translational error (m) and rotational error (°). Numbers in **bold** represent the best performance.



(a) dSLAM NeRF (b) SfM NeRF

Figure 4. Qualitative comparison between the NeRFs trained by dSLAM GT pose (a) vs. SfM GT pose (b). As illustrated, SfM NeRF (PSNR 19.94 dB) can render superior geometric details (bottom row) than dSLAM NeRF (PSNR 16.11 dB).

4.2. Evaluation on 7-Scenes

We further evaluate our method on Microsoft 7-Scenes dataset [19, 55], which includes seven indoor scenes ranging in size from $1m^3$ to $18m^3$ and up to 7000 training images for each scene. We sub-sampled the large training sequences in the dataset to 1000 images for training our NeFeS model. The original 'ground truth' (GT) poses are obtained from RGB-D SLAM (dSLAM) [38]. However, we observe imperfection in the GT poses due to the asynchronous data between the RGB and depth sequences, and this results in low-quality NeRF renderings, as shown in Fig. 4. Thus, we use alternative 'ground truth' provided by [7] for our experiments. The authors [7] demonstrate that their camera poses reconstructed by COLMAP [50], a structure-from-Motion (SfM) library, are more accurate for image-based relocalization. We refer the reader to our supplementary [4] for more details about the GT poses.

For a fair comparison, we use the SfM poses to re-train baseline APR methods using their official code, except for PoseNet, in which we use the open-sourced code from [10]. We trained each APR method 3-4 times to select the best-performing model. We use DFNet + NeFeS$_{50}$ with the same settings as in Cambridge for our pose refinement experiment. We achieve state-of-the-art results (59%+ better in scene average) in all scenes by running 50 optimization steps (see Tab. 2). The results by using the dSLAM ground-truth poses are included in the supplementary[5], where we also achieve SOTA accuracy.

It is imperative to note that our method is not constrained by the number of optimization steps for the refinement process. In our experience, nearly 50% of the entire pose error improvement is accomplished within the initial 10 steps. It

[4] Supplementary: 7-Scenes Dataset Details
[5] Supplementary: APR Comparisons with dSLAM GT

Dataset	7-Scenes	Cambridge
PoseNet	0.21m/7.74	2.04m/6.23
+ Ours	**0.08m/2.83°**	**0.54m/1.05°**
MS-Trans.	0.18m/9.51°	1.28m/2.73°
+ Ours	**0.11m/3.46°**	**0.43m/1.04°**
DFNet	0.06m/1.93°	1.19m/2.90°
+ Ours	**0.02m/0.79°**	**0.35m/0.77°**

Table 3. **Pose refinement on different APR architectures.** Our refinement method can effectively improve pose estimation results for different APR architectures.

Dataset	NetVlad	Optimize Pose
7-Scenes	0.32m/13.72°	**0.14m/3.97°**
Cambridge	3.18m/7.74°	**1.15m/1.30°**

Table 4. **Pose refinement on NetVlad** Our method also works on poses initialized with non-APR-based methods, such as NetVlad image retrieval. Since the initial pose error is relatively large, we refine the poses with 100 iterations.

is up to the user to find the optimal trade-off that fits their computational budget. A detailed analysis of this facet is provided in Sec. 4.7.

4.3. Refinement for Different APRs

Tab. 3 shows the results of our method with different APR architectures. Our proposed method exhibits versatility, operating beneficially under various APR architectures, such as PoseNet (classic pose regression architecture), MS-Transformer (EfficentNet CNN backbones with transformer blocks), and DFNet (multi-task network that predicts domain invariant features and poses). A full table with per-scene results is provided in Supplementary.

4.4. Optimize APR vs. Optimize Pose

Besides boosting APR, our proposed approach can also refine the initial coarse camera pose, as outlined in Sec. 3.3. We first show a use case of this scenario by coupling our method with image retrieval, where the initial pose can only be optimized due to the non-differentiable nature of the retrieval process. Given a query image, we retrieve its nearest neighbor from the training data using NetVLAD [1] and use the associated pose as the initial pose. We set the learning rate to be lr_R and lr_t for rotation and translation components, respectively. Specifically, for indoor scenes, we set $lr_R = 0.0087$ (corresponds to $0.5°$ in radiance) and $lr_t = 0.01$. For outdoor scenes, we set $lr_R = 0.01$ and $lr_t = 0.1$. Tab. 4 summarises the experimental results, indicating substantial improvements in pose accuracy over the NetVLAD retrieved coarse pose, exceeding the performance of many prior APR approaches.

Dataset	DFNet	Optimize Pose	Optimize APR
7-Scenes	0.06m/1.93°	0.04m/1.16°	**0.02m/0.79°**
Cambridge	1.19m/2.90°	0.66m/1.39°	**0.35m/0.77°**

Table 5. **Pose refinement vs. APR refinement** We study on the optimization over APR vs. the pose. Both methods can effectively optimize poses. However, optimizing APR can obtain lower errors than optimizing poses given the same number of iterations.

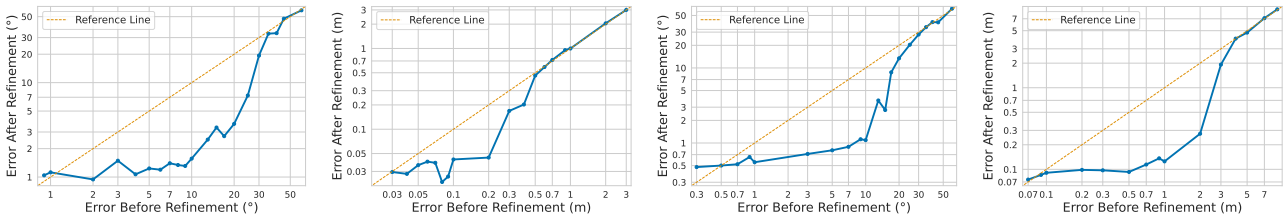
We further conducted controlled experiments to investigate if performance disparities exist between two types of optimization: optimizing the APR's parameters or directly optimizing the pose itself. We evaluated both modes on the DFNet with NeFeS_{50} refinement, as illustrated in Tab. 5. The result suggests that while both refinement approaches can effectively improve the pose accuracy, optimizing the neural network's parameters obtains a better result than directly optimizing the pose itself, which is an interesting insight. Nevertheless, optimizing the pose remains is also valuable, particularly when the initial pose is derived from a non-differentiable or a black-box pose estimator.

4.5. Pose Refinement Bounds

Our refinement method relies on matching rendered features with query image features during test time, so it may fail when there is not sufficient overlap between the two feature maps. To determine the bounds of our refinement method, we randomly perturb the ground-truth pose and determine the maximum perturbation at which our method stops converging. We jitter the orientation or position of the ground truth pose components separately while gradually increasing the magnitude of the perturbation. We use two scenes, an indoor scene (*7-Scenes: Heads*) and an outdoor scene (*Cambridge: Shop Facade*), to illustrate our results in Fig. 5. We observe that our method cannot refine pose errors larger than $35°$. In case of translational errors, our method can refine errors up to $0.6m$ on *Heads* (indoor scene) and up to $4m$ in *Shop Facade* (outdoor scene). This difference may come from the discrepancy in scale between indoor and outdoor settings. For example, in the small-scale scene of *Heads*, the camera is closer to the objects, hence even small movements lead to a large change in the rendering.

4.6. NeFeS Ablation

This section presents the ablation study of our NeFeS network. In Tab. 6a, we gradually remove the exposure-adaptive ACT and the Feature Fusion module and evaluate their impact on the performance of our approach on *Cambridge Shop Facade*. The results demonstrate that the removal of each component leads to a deterioration in pose accuracy, indicating the effectiveness of both components.



(a) Rotation test (Indoor) (b) Translation test (Indoor) (c) Rotation test (Outdoor) (d) Translation test (Outdoor)

Figure 5. Experiments on pose refinement bounds of our method in indoor and outdoor scenes. Each plot shows errors before (x-axis) and after (y-axis) refinement when ground-truth pose is perturbed by varying magnitudes. Dashed green line is '$y{=}x$'. Points below this line indicate a reduction in pose error using our refinement method.

(a) NeFeS Architecture Ablation

Method	Shop Facade
Initial Pose Error	0.67m/2.21°
Refine w/ NeFeS (ours)	**0.15m/0.53°**
- Exposure-adaptive ACT Ⓐ	0.15m/1.20°
- Ⓐ+Feature Fusion	0.37m/1.62°

(b) Training Scheduling Ablation

Method	Shop Facade
Combined	0.17m/0.80°
Progressive	**0.15m/0.53°**

Table 6. **(a)** Ablation on NeFeS architecture. **(b)** Ablation on the proposed training scheduling

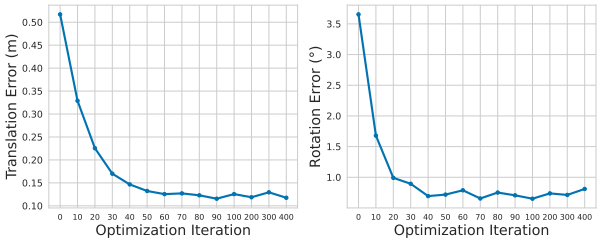


Figure 6. Plots of rotation and translation errors against the number of iteration on *Cambridge: Shop Facade* scene.

A noteworthy insight from our architecture design is that the superior pose estimation accuracy is attributed to integrating both Exposure-adaptive ACT and the Feature Fusion module. The feature fusion module can smooth potential noises (outliers) from directly-rendered 3D features.

In Tab. 6b, we compare our progressive training scheduling with the combined scheduling, where all three loss terms have been enabled simultaneously since the beginning of the training. The results reveal that the progressive training scheduling results in better accuracy, providing further support for our design decisions.

4.7. Number of Iterations vs. Accuracy Trade-off

In Fig. 6, we plot the relationship between the number of optimization iterations and pose error. Both translation and rotation errors reduce significantly within the first 10-20 iterations. Hence, given a computational budget, an operating point can be (optionally) chosen w.r.t. the performance-time trade-off. The errors start to plateau around 50 steps. Although we can achieve even lower errors with more iterations, we think using 50 steps strikes a balance between accuracy and efficiency, and explains how we set our previous experiments.

4.8. Spatial vs Channel-wise Normalization

As described in Sec. 3.1, we empirically find spatial-wise normalized features yield higher accuracy than channel-wise normalized features when computing the feature cosine similarity loss $\mathcal{L}_{feature}$, evidenced by Tab. 7. This is likely due to our spatially sensitive dense direct matching, akin to methods like Direct Sparse Odometry [18]. In contrast, channel-wise normalization can introduce inconsistencies among neighboring pixels.

Methods	Shop
Channel	0.20m/1.05°
Spatial	**0.15m/0.53°**

Table 7. Performance comparison between using Channel-wise vs. Spatial-wise normalization in loss $\mathcal{L}_{feature}$ during refinement.

5. Conclusion

We tackle the camera relocalization problem and improve absolute pose regression (APR) methods by proposing a test-time refinement method. In particular, we design a novel model named Neural Feature Synthesizer (NFS), which can encode 3D geometric features. Given an estimated pose, the NFS renders a dense feature map, compares it with the query image features, and back-propagates the error to refine estimated camera poses from APR methods. In addition, we propose a progressive learning strategy and a feature fusion module to improve the feature robustness of the NFS model. The experiments demonstrate that our method can greatly improve the accuracy of APR methods. Our method provides a promising direction for improving the accuracy of APR methods.

6. Supplementary

6.1. Implementation Details

6.1.1 Architecture details

The model is trained with the re-aligned and re-centred poses in SE(3), as in [32]. We use a coarse-to-fine sampling strategy with 64 sampled points per ray in both stages. The width of the MLP layers is 128 and we output $N_c = 3$ and $N_f = 128$ in the last layer of the fine stage MLP. For the exposure-adaptive ACT module, we compute the query image's histogram $\mathbf{y}_I$ in YUV color space and bin the luminance channel into $N_b = 10$ bins. We then feed the binned histogram to 4-layer MLPs with a width of 32. The exposure-adaptive ACT module outputs the exposure compensation matrix $\mathbf{K}$ and the bias $\mathbf{b}$, which directly transform the integrated colors $\hat{\mathbf{C}}_{NFS}(\mathbf{r})$ of the main networks, with negligible computational overhead. We run the APR refinement process for m iterations per image using the direct feature matching loss $\mathcal{L}_{feature}$ with a learning rate of 1×10^{-5}. Our default value for m is 50 unless specified, denoted as NeFeS$_{50}$. The NeFeS model renders features with a shorter side of 60 pixels and then upsample them using bicubic interpolation to 240 for feature matching.

6.1.2 Progressive training schedule

The training process for the NeFeS network starts with the photometric loss only for $T_1 = 600$ epochs by setting $\lambda_1 = \lambda_2 = 0$ in Eq. (4). The color and density components of the model are trained with a learning rate of 5×10^{-4} which is exponentially decayed to 8×10^{-5} over 600 epochs. We randomly sample 1536 rays per image and use a batch size of 4. After 600 epochs, we reset the learning rate to 5×10^{-4} and switch on the feature loss ($\mathcal{L}_f$ in Eq. (6)) for the next $T_2 = 200$ epochs with $\lambda_1 = 0.04, \lambda_2 = 0$. The fusion loss ($\mathcal{L}_{fusion}$ in Eq. (7)) is switched on for the last $T_3 = 400$ epochs with coefficients $\lambda_1 = 0.02, \lambda_2 = 0.02$. During the third training stage T_3, instead of randomly sampling image rays, we randomly sample $N_{crop} = 7$ image patches of size $S \times S$ where $S = 16$. To extract image features (i.e. $\mathbf{F}_{\mathbf{img}}(I, \cdot)$) as pseudo-groundtruth, we use the finest-level features from DFNet's [11] feature extractor module. We resize the shorter sides of the feature labels to 60.

6.2. Refinement for Different APRs Full

This is the supplementary full table for Section 4.3 of the main paper (Tab. 8).

6.3. Qualitative Comparisons

In Fig. 8, we qualitatively compare the refinement accuracy of different APR methods - namely PoseNet[21–23], MS-Transformer[54], DFNet [11] - with our method, i.e. DFNet+NeFeS$_{50}$. We can observe that our method produces the most accurate poses (compared to ground-truth) and has a significant improvement over DFNet in different scenes such as fire [1000-1500] and kitchen [1000-1500].

Dataset	PoseNet	+ Ours	MS-Trans.	+ Ours	DFNet	+ Ours
7-Scenes	pose error in m/°					
Chess	0.10/4.02	0.04/1.35	0.11/6.38	0.06/1.96	0.03/1.12	0.02/0.57
Fire	0.27/10.0	0.03/1.20	0.23/11.5	0.06/2.55	0.06/2.30	0.02/0.74
Heads	0.18/13.0	0.12/7.91	0.13/13.0	0.09/6.19	0.04/2.29	0.02/1.28
Office	0.17/5.97	0.02/0.72	0.18/8.14	0.05/1.69	0.06/1.54	0.02/0.56
Pumpkin	0.19/4.67	0.06/1.57	0.17/8.42	0.07/1.85	0.07/1.92	0.02/0.55
Kitchen	0.22/5.91	0.02/0.68	0.16/8.92	0.08/2.31	0.07/1.74	0.02/0.57
Stairs	0.35/10.5	0.27/6.35	0.29/10.3	0.34/7.64	0.12/2.63	0.05/1.28
Average	0.21/7.74	**0.08/2.83**	0.18/9.51	**0.11/3.46**	0.06/1.93	**0.02/0.79**
Cambridge	pose error in m/°					
Kings	1.66/4.86	0.38/0.56	0.83/1.47	0.43/0.59	0.73/2.37	0.37/0.54
Hospital	2.62/4.90	1.15/1.30	1.81/2.39	0.61/1.06	2.00/2.98	0.52/0.88
Shop	1.41/7.18	0.21/0.81	0.86/3.07	0.18/0.98	0.67/2.21	0.15/0.53
Church	2.45/7.96	0.42/1.52	1.62/3.99	0.48/1.53	1.37/4.03	0.37/1.14
Average	2.04/6.23	**0.54/1.05**	1.28/2.73	**0.43/1.04**	1.19/2.90	**0.35/0.77**

Table 8. **Pose refinement on different APR architectures.** Our refinement method can effectively improve pose estimation results for different APR methods. PoseNet is the classic pose regression architecture. MS-Transformer is denoted as MS-Trans., which combines EfficentNet CNN backbones with transformer blocks. DFNet is a multi-task network that predicts domain invariant features and poses.

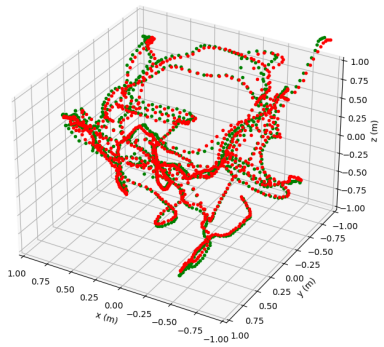
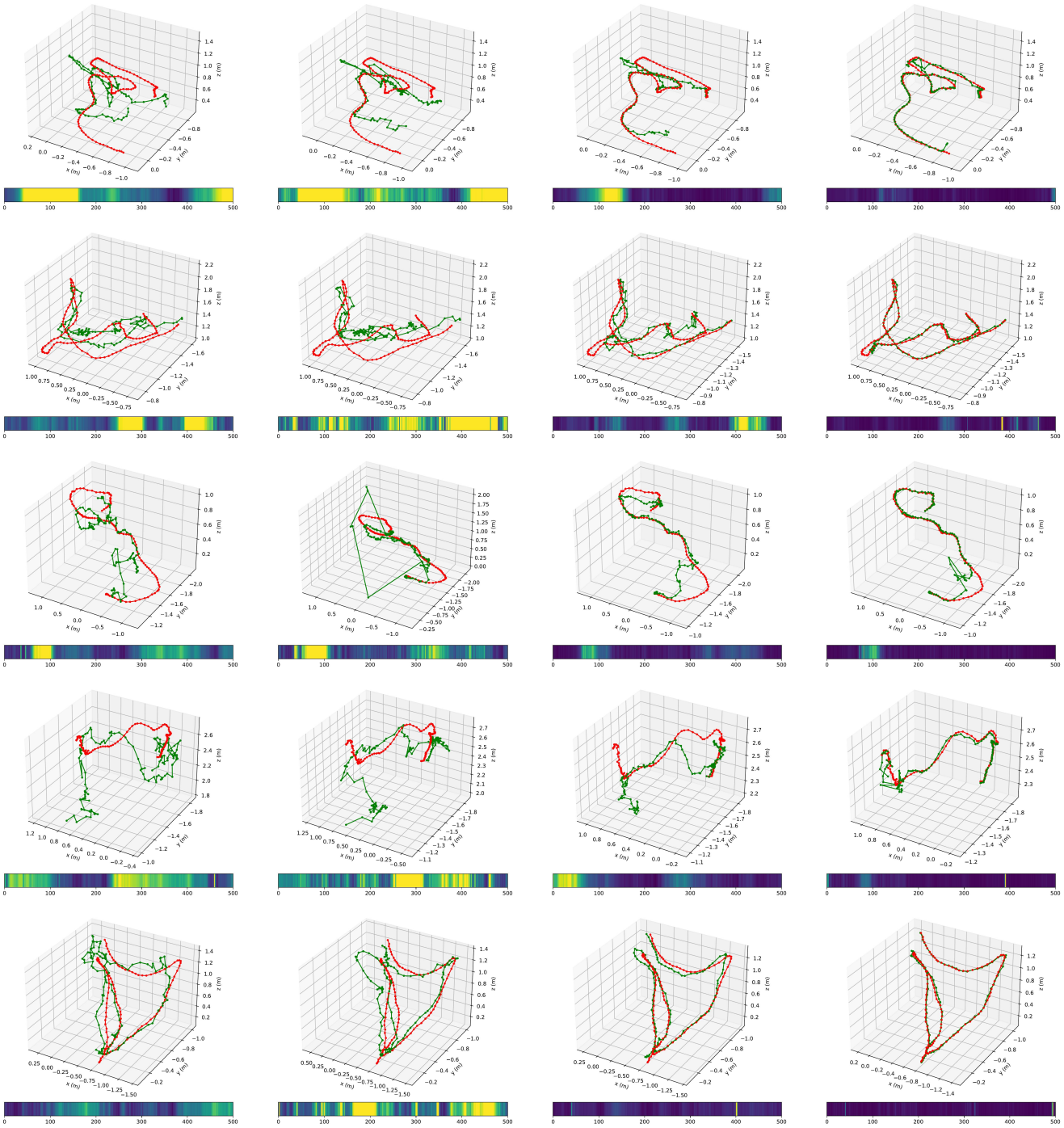


Figure 7. A visualization of camera trajectories of *7-Scene: Chess* scene. The original 'GT' poses are obtained using dSLAM [38] (green). In this paper, we use SfM GT poses provided by [7] (red) for better GT pose accuracy. Two GT trajectories have a median ATE error of 3.5cm/1.46°.

6.4. 7-Scenes Dataset Details

In Sec. 4.2 of the main paper, we mention the difference between the dSLAM-generated ground-truth pose and the SfM-generated ground-truth pose for the 7-Scenes dataset. We provide more details in this section.

dSLAM vs. SfM GT pose Brachmann *et al.* [7] identified imperfections in the original 'ground-truth' (GT) poses generated by dSLAM in the 7-Scenes dataset. The erro-



(a) PoseNet (b) MS-Transformer (c) DFNet (d) **DFNet+NeFeS**$_{50}$

Figure 8. Qualitative comparison on the 7-Scenes dataset. The 3D plots show the camera positions: green for ground truth and red for predictions. The bottom color bar represents rotational errors for each subplot, where yellow means large error and blue means small error for each test sequence. Sequence names from top to bottom are: fire [1000-1500], office [2500-3000], pumpkin [500-1000], kitchen [1000-1500], kitchen [1500-2000]. Since each scene has different numbers of frame, we select 500 frames from each of them and append the range after scene's name.

neous GT poses originate from sensor asynchronization between the captured RGB images and depth maps. Therefore, Brachmann et al. employed SfM to regenerate a new set of 'ground-truth' poses, which subsequently aligned and scaled to match the dSLAM-derived poses. As described in Sec. 4.2 of the main paper, we notice that when trained with the SfM ground-truth poses, the rendering quality of NeRF is noticeably boosted compared with using the dSLAM GT poses. The comparison between the trajectories of two sets of ground-truth poses is visualized in Fig. 7. An interesting observation is made based on the results presented in Table 2 of our main paper. We notice DFNet achieves superior performance when trained with SfM-grounded GT data, surpassing its performance as originally reported [11]. This phenomenon may be attributed to utilizing the improved synthetic dataset generated by NeRF during DFNet's *Random View Synthesis* training.

APR Comparisons with dSLAM GT. To supplement Table 2 of the main paper, we compare previous methods and our method when trained and evaluated using dSLAM GT poses. The results can be found in Tab. 9. Note that the pose error is presented in cm/degree to emphasize the distinctions in translational accuracy. Despite NeFeS models being trained using suboptimal dSLAM GT poses in this experiment which reduces the quality of the feature rendering, our model is able to achieve SOTA performance on single-frame APR comparisons. Notably, Coordinet+LENS [33] is the only single-frame APR technique that achieves our method's proximate outcomes (on translational error). However, it's pertinent to note that LENS requires several days to train a high-quality NeRF model per scene. In stark contrast, the NeFeS model requires a much shorter training duration of approximately 5-20 hours, accompanied by an inference speed over 110 times faster and obviated the need for manual parameter tuning, making NeFeS a notably more cost-effective prospect.

Furthermore, we experiment to see if the current dSLAM pose results can be improved if a better quality NeFeS model is used. We performed joint optimization of NeFeS and ground truth camera poses during training using the method introduced in NeRF– – [65]. The outcomes reveal that while the NeFeS model attains an enhanced training PSNR from 23.33dB to 27.88dB and the median translation error improves by $1cm$, the rotation error worsens by $0.07°$ since jointly optimizing the dSLAM GT training poses also slightly shifts the world coordinate system of the radiance fields. This refined model's performance is denoted as DFNet + NeFeS$_{50}^{--}$, as indicated in Tab. 9.

Methods	Average(cm/°)
PoseNet(PN)[23]	44/10.4
PN Learn σ^2[22]	24/7.87
geo. PN[22]	23/8.12
LSTM PN[63]	31/9.85
Hourglass PN[31]	23/9.53
BranchNet[66]	29/8.30
MapNet[8]	21/7.77
Direct-PN[10]	20/7.26
TransPoseNet[53]	18/7.78
MS-Transformer[54]	18/7.28
MS-Transformer+PAE [52]	15/7.28
CoordiNet[34]	22/9.7
CoordiNet+LENS[33]	8/3.0
DFNet [11]	12/3.71
DFNet + NeFeS$_{50}$ (dSLAM)	8/**2.80**
DFNet + NeFeS$_{50}^{--}$ (dSLAM)	**7**/2.87

Table 9. We compare the proposed refinement method using 7-Scene dSLAM GT pose [55] with prior single-frame APR methods, in average of median **translation error (cm)** and **rotation error** (°). Numbers in **bold** represent the best performance.

6.5. Comparison with Other Camera Localization Approaches

Although our paper mainly focuses on test-time refinement on single-frame APR methods, it is only one family of approaches in camera relocalization (see our Related Work section). In Tab. 10, we compare geometry-based methods and sequential-based methods for camera localization, as well as adding several other single-frame APR methods, including some without code available publicly to support a more thorough comparison. The results on 7-Scenes dataset are evaluated using the original SLAM ground-truth pose, except methods marked by "(COLMAP)", which indicates the results evaluated using the COLMAP ground-truth pose for 7-Scenes. The methods that marked by "(COLMAP to build 3D model)" indicates COLMAP generated 3D models are used in training and evaluation.

We show that when compared with sequential-based APR methods, our method achieves very competitive results on Cambridge Landmark dataset and 7-Scenes dataset. In addition, for the first time, we show that a single-frame APR method can obtain accuracy of the same magnitude as 3D geometry-based approaches.

6.6. Featuremetric vs. Photometric Refinement

In this section, we study the differences between featuremetric refinement and photometric refinement. Prior literature such as iNeRF [67], NeRF−− [65], BARF [27], GARF [14], and NoPe-NeRF [3], have attempted to 'invert' a NeRF model with photometric loss for pose optimization.

However, directly comparing our featuremetric method with these methods would not be appropriate due to the following reasons: **Firstly**, these methods [3, 14, 27, 65] optimize both camera and NeRF model parameters simultaneously but are unsuitable for complex scenes with large motion (e.g. 360°scenes) since each frame's camera pose is

Family	Method	Cambridge	7-Scenes
Seq. 3D	KFNet[68]	13/0.3	3/0.88
3D	AS[47]	29/0.6	5/2.5
	AS[48]	11/0.3	4/1.2
	DSAC[6]	32/0.8	20/6.3
	DSAC*[5]	15/0.4	3/1.4
	DSAC*[7] (COLMAP)	-	1/0.34
	PixLoc[46] (COLMAP to build 3D model)	11/0.3	3/1.1
	HLoc [44] (COLMAP to build 3D model)	10/0.2	3/1.09
Seq. APR	MapNet+PGO[8]	-	18/6.55
	AtLoc+[64]	-	19/7.08
	TransAPR[41]	94/2.12	17/6.29
	VLocNet [61]	78/2.82	5/3.80
1-frame APR	PoseNet(PN)[23]	204/6.23	44/10.4
	PN Learn σ^2[22]	143/2.85	24/7.87
	geo. PN[22]	163/2.86	23/8.12
	LSTM PN[63]	130/5.51	31/9.85
	Hourglass PN[31]	-	23/9.53
	BranchNet[66]	-	29/8.30
	MapNet[8]	163/3.64	21/7.77
	Direct-PN[10]	-	20/7.26
	TransPoseNet[53]	91/3.50	18/7.78
	MS-Transformer[54]	128/2.73	18/7.28
	MS-Transformer+PAE [52]	96/2.73	15/7.28
	E-PoseNet [36]	94/2.12	17/7/32
	CoordiNet[34]	92/2.58	22/9.7
	CoordiNet+LENS[33]	39/1.15	8/3.0
	DFNet [11]	119/2.90	12/3.71
	DFNet [11] (COLMAP)	-	6/1.93
	DFNet + NeFeS$_{50}$	**35/0.77**	**8/2.80**
	DFNet + NeFeS$_{50}$ (COLMAP)	-	**2/0.79**

Table 10. This table compares different types of camera relocalization on Cambridge Landmarks and 7-Scenes dataset. We show representative methods for each school of approach: geometry-based methods (3D), sequential-based APR methods (Seq. APR), and single-frame APR methods (1-frame APR). We report the average of median **translation error (cm)** and **rotation error** (°). Numbers in **bold** represent the performance of our methods.

initialized from an identity matrix. **Secondly**, these methods do not effectively handle exposure variations, resulting in suboptimal rendering quality. **Thirdly**, even with a coarse camera pose initialization, photometric-based inversion methods cannot prevent drifting in refined camera poses, leading to misalignment with the ground truth poses of testing sequences.

Therefore, for a fair comparison with photometric methods, we define a photometric refinement model as the baseline model to compare with. Specifically, for the baseline model, the main architecture from the NeFeS model is maintained but without the feature outputs, and only the RGB colors $\hat{\mathbf{C}}(\mathbf{r})$ are used for photometric pose refinement. The performance of two cases with photometric refinement are demonstrated in Tab. 11: first is a sparse photometric refinement that randomly samples pixel-rays, similar to iNeRF [67] or BARF[27]-like methods; and the other uses dense photometric refinement, which renders entire RGB images. The results indicate that our featuremetric refinement is more robust than all the photometric refinement baselines, as it achieves lower pose errors after 50 iterations

Methods	Hospital
DFNet	2.00m/2.98°
DFNet + Sparse NeRF photometric$_{50}$	1.19m/1.52°
DFNet + Dense NeRF photometric$_{50}$	0.80m/1.12°
DFNet + NeFeS$_{50}$	**0.52m/0.88°**

Table 11. We compare our featuremetric refinement method using the proposed **NeFeS** network with photometric-based refinement baselines on *Cambridge Hospital*.

	LR Settings	Shop-20% (+NeFeS$_{20}$)
Same lr	Initial Pose Error	0.58m/3.14°
	$lr_R = lr_t = 0.1$	0.91m/22.70°
	$lr_R = lr_t = 0.01$	0.49m/1.51°
	$lr_R = lr_t = 0.003$	0.54m/2.44°
	$lr_R = lr_t = 0.001$	0.57m/2.48°
Different lr	$lr_R = 0.01, lr_t = 0.1$	**0.27m/1.77°**

Table 12. We use a toy example to show the benefit of using *different* learning rates over *same* learning rates for translation and rotation components during direct pose refinement. We show four cases for *same* learning rate including two settings that are used in prior works. Our pose refinement results are evaluated by using 20% test data of *Cambridge: Shop Facade* and 20 iterations of optimization.

of optimization.

6.7. Benefit of splitting lr_R and lr_t

As described in Sec. 3.3 of the main paper, we find using different learning rates for translation and rotation components as beneficial for fast convergence when we directly refine the camera pose parameters. In this section, we use a toy experiment to illustrate how we determine to use this strategy. We select 20% of *Cambridge: Shop Facade*'s test images and perform direct pose refinement for 20 iterations using our NeFeS model. In Tab. 12, we compare our *different* learning rate setting with several cases of *same* learning rate settings. The learning rate $lr_R = lr_t = 0.003$ is used in [14, 27] and $lr_R = lr_t = 0.001$ is used in [65, 67]. We show that by utilizing a different learning rate strategy, the pose error converges much faster and is more stable for both camera position and orientation.

6.8. Runtime Analysis

Runtime cost. Due to better implementation flexibility, we used an unoptimized version of NeFeS in this study. The pytorch-based NeFeS currently runs at 6.9 fps per image including its backpropagation, which is 3x faster than DFNet's NeRF-Hist [11] and 110x faster than LENS's NeRF-W [33]. It is crucial to emphasize that further optimization can be pursued to attain commercial-level effi-

ciency. For example, NeFeS can potentially be accelerated up to 66x using the C++/CUDA-based `tiny-cuda-nn` and `instant-ngp` [35] frameworks.

Training cost. Our NeFeS can be trained in parallel with the APR method such as DFNet and takes roughly the same time as the underlying APR method (*i.e.* 5-20 hrs depending on scene size). However, the NeFeS model only needs to be trained **once** and the same model can be applied to different APR methods.

6.9. Additional Insight

DFNet features vs. other type of features. We were curious about how NeFeS performs when trained with features other than the DFNet. Thus, we experimented with training the NeFeS model with PixLoc [46] features in our refinement pipeline. While we did find positive results, the refined performance didn't reach that of DFNet features. This is because DFNet is trained to close the domain gap between features extracted from natural query images and features rendered by NeRF.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016. 7

[2] Yash Bhalgat, Iro Laina, João F Henriques, Andrew Zisserman, and Andrea Vedaldi. Contrastive lift: 3d object instance segmentation by slow-fast contrastive fusion. In *NeurIPS*, 2023. 2

[3] Wenjing Bian, Zirui Wang, Kejie Li, Jiawang Bian, and Victor Adrian Prisacariu. NoPe-NeRF: Optimising neural radiance field with no pose prior. In *CVPR*, 2023. 2, 5, 11

[4] Eric Brachmann and Carsten Rother. Learning Less is More - 6D Camera Localization via 3D Surface Regression. In *CVPR*, 2018. 2

[5] Eric Brachmann and Carsten Rother. Visual camera relocalization from RGB and RGB-D images using DSAC. *IEEE TPAMI*, 2021. 1, 12

[6] 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, 12

[7] 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, 9, 12

[8] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-Aware Learning of Maps for Camera Localization. In *CVPR*, 2018. 1, 2, 6, 11, 12

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2

[10] Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-PoseNet: Absolute pose regression with photometric consistency. In *3DV*, 2021. 1, 2, 4, 6, 11, 12

[11] Shuai Chen, Xinghui Li, Zirui Wang, and Victor Prisacariu. DFNet: Enhance absolute pose regression with direct feature matching. In *ECCV*, 2022. 1, 2, 3, 4, 5, 6, 9, 11, 12

[12] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 2

[13] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 5

[14] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. Garf: Gaussian activated radiance fields for high fidelity reconstruction and pose estimation. In *ECCV*, 2022. 2, 5, 11, 12

[15] Ronald Clark, Sen Wang, Andrew Markham, Niki Trigoni, and Hongkai Wen. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In *CVPR*, 2017. 2

[16] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. 2

[17] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features. In *CVPR*, 2019. 2

[18] J. Engel, V. Koltun, and D. Cremers. DSO: Direct sparse odometry. In *IEEE TPAMI*, 2017. 8

[19] Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time rgb-d camera relocalization. In *ISMAR*, 2013. 6

[20] M. Irani and P. Anandan. All about direct methods. In *Workshop Vis. Algorithms: Theory Pract.*, 1999. 4

[21] A. Kendall and R. Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *ICRA*, 2016. 1, 2, 9

[22] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017. 6, 11, 12

[23] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015. 1, 2, 5, 6, 9, 11, 12

[24] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. In *NeurIPS*, 2022. 2, 4

[25] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022. 2

[26] Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-resolution correspondence networks. In *NeurIPS*, 2020. 2, 3

[27] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021. 2, 5, 11, 12

[28] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-Perfect Structure-from-Motion with Featuremetric Refinement. *ICCV*, 2021. 2

[29] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duck-

worth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021. 1, 2, 3, 4

[30] N. Max. Optical models for direct volume rendering. In *IEEE Transactions on Visualization and Computer Graphics*, 1995. 3, 4

[31] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu. Image-based localization using hourglass networks. In *ICCVW*, 2017. 2, 11, 12

[32] 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. 1, 2, 3, 9

[33] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. LENS: Localization enhanced by nerf synthesis. In *CoRL*, 2021. 1, 2, 11, 12

[34] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Coordinet: uncertainty-aware pose regressor for reliable vehicle localization. In *WACV*, 2022. 1, 2, 11, 12

[35] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 2022. 13

[36] Mohamed Adel Musallam, Vincent Gaudillie‘re, Miguel Ortiz del Castillo, Kassem Al Ismaeil, and Djamila Aouada. Leveraging equivariant features for absolute pose regression. In *CVPR*, 2022. 12

[37] T. Naseer and W. Burgard. Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In *IROS*, 2017. 2

[38] R. Newcombe, A. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011. 6, 9

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019. 5

[40] Pulak Purkait, Cheng Zhao, and Christopher Zach. Synthetic view generation for absolute pose regression and image synthesis. In *BMVC*, 2018. 2

[41] Chengyu Qiao, Zhiyu Xiang, Yuangang Fan, Tingming Bai, Xijun Zhao, and Jingyun Fu. TransAPR: Absolute camera pose regression with spatial and temporal attention. In *IEEE Robotics and Automation Letters*, 2023. 12

[42] N. Radwan, A. Valada, and W. Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. In *IEEE Robotics and Automation Letters*, 2018. 2

[43] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *CVPR*, 2022. 4

[44] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 2, 12

[45] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2

[46] Paul-Edouard Sarlin, Ajaykumar Unagar, Måns Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, and Torsten Sattler. Back to the Feature: Learning robust camera localization from pixels to pose. In *CVPR*, 2021. 1, 2, 12, 13

[47] T. Sattler, B. Leibe, and L. Kobbelt. Improving image-based localization by active correspondence search. In *ECCV*, 2012. 12

[48] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. In *IEEE TPAMI*, 2017. 1, 2, 12

[49] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *CVPR*, 2019. 1, 3

[50] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 6

[51] Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. In *Workshop on Language and Robotics at CoRL*, 2022. 2

[52] Yoli Shavit and Yosi Keller. Camera pose auto-encoders for improving pose regression. In *ECCV*, 2022. 1, 2, 6, 11, 12

[53] Yoli Shavit, Ron Ferens, and Yosi Keller. Paying attention to activation maps in camera pose regression. In *arXiv preprint arXiv:2103.11477*, 2021. 2, 6, 11, 12

[54] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *ICCV*, 2021. 2, 6, 9, 11, 12

[55] 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. 6, 11

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

[57] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021. 2, 3

[58] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, , and Akihiko Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. *CVPR*, 2018. 2

[59] Zachary Teed and Jia Deng. Tangent space backpropagation for 3d transformation groups. In *CVPR*, 2021. 5

[60] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural Feature Fusion Fields: 3D distillation of self-supervised 2D image representations. In *3DV*, 2022. 2

[61] A. Valada, N. Radwan, and W. Burgard. Deep auxiliary learning for visual localization and odometry. In *ICRA*, 2018. 2, 12

[62] 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. 3

[63] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-based localization using lstms for structured feature correlation. In *ICCV*, 2017. 2, 6, 11, 12

[64] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. In *AAAI*, 2020. 2, 12

[65] 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. 2, 5, 11, 12

[66] J. Wu, L. Ma, and X. Hu. Delving Deeper into Convolutional Neural Networks for Camera Relocalization. In *ICRA*, 2017. 2, 11, 12

[67] 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 *arxiv arXiv:2012.05877*, 2020. 2, 11, 12

[68] Lei Zhou, Zixin Luo, Tianwei Shen, Jiahui Zhang, Mingmin Zhen, Yao Yao, Tian Fang, and Long Quan. Kfnet: Learning temporal camera relocalization using kalman filtering. In *CVPR*, 2020. 12

[69] 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

Statement of Authorship for the paper "Neural Refinement for Absolute Pose Regression with Feature Synthesis" in Chapter 5.

Paper title	Neural Refinement for Absolute Pose Regression with Feature Synthesis
Authors	**Shuai Chen**, Yash Bhalgat, Xinghui Li, Jia-Wang Bian, Kejie Li, Zirui Wang, Victor Adrian Prisacariu
Publication status	Published
Publication details	Conference on Computer Vision and Pattern Recognition (CVPR), 2024.

Student Confirmation

Student name	Shuai Chen	
Contribution to the paper	First-author contribution: • Conception of research ideas • Design and implementation of models • Running of large-scale experiments • Writing and presentation of the paper	
Signature and Date		Apr. 23th 2025

Supervisor Confirmation

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

Supervisor name	Prof. Victor Adrian Prisacariu	
Supervisor comments	The description is accurate	
Signature and Date		Apr. 23th 2025

Chapter 6

Hierarchical Pose Refinement via Uncertainty Estimation

This chapter presents a follow-up study that addresses the limitations of the neural pose refinement method introduced in Chapter 5. While NeRF-based refinement methods such as NeFeS improve pose accuracy by optimising over learnt neural feature fields, they typically rely on a time-consuming iterative approach that requires numerous refinement steps. This strategy significantly increases computational latency to obtain the optimal performance, limiting the efficiency of pose regression methods. Enhancing runtime performance, particularly for methods such as NeFeS, is therefore crucial for practical applicability.

This work proposes Hierarchical Refinement-based APR (HR-APR), an efficient extension of NeFeS that introduces uncertainty-aware refinement and a hierarchical optimisation strategy to reduce test-time computational cost. Central to this framework is a lightweight uncertainty estimation module that predicts pose reliability at test time by computing the cosine similarity between query features and training set embeddings. It is important to clarify that this measure serves as a similarity-based proxy for uncertainty rather than representing a full proba-

bilistic distribution. This choice is empirically justified by its strong correlation with pose error, as higher feature similarity typically indicates that the query is well-represented by the training distribution. Unlike previous uncertainty-based methods, our design for the uncertainty estimation module is lightweight (typically under 7 ms in a standard consumer grade GPU), requires no additional training labels, and can be seamlessly integrated into existing APR architectures.

Moreover, the hierarchical pipeline built atop NeFeS dynamically allocates refinement efforts based on these uncertainty scores, prioritising optimisation for less reliable poses while efficiently handling high-confidence predictions. Conceptually, this mechanism is close in spirit to traditional place-recognition approaches, which rely on retrieving similar scene content to localise a camera. However, while standard place-recognition pipelines often serve as a coarse indexing step, our approach utilises feature similarity as a continuous proxy to bridge the gap between absolute pose regression and local coordinate-level refinement.

Our experiment shows that using this hierarchical pose refinement can reduce the computational overhead of NeFeS by 27.4% and 15.2% on the indoor and outdoor dataset. This design reduces computational overhead while maintaining or improving pose accuracy.

This work was accepted at the *2024 International Conference on Robotics and Automation*.

HR-APR: APR-agnostic Framework with Uncertainty Estimation and Hierarchical Refinement for Camera Relocalisation

Changkun Liu[1], Shuai Chen[3], Yukun Zhao[1], Huajian Huang[1], Victor Prisacariu[3] and Tristan Braud[1,2]

***Abstract*— Absolute Pose Regressors (APRs) directly estimate camera poses from monocular images, but their accuracy is unstable for different queries. Uncertainty-aware APRs provide uncertainty information on the estimated pose, alleviating the impact of these unreliable predictions. However, existing uncertainty modelling techniques are often coupled with a specific APR architecture, resulting in suboptimal performance compared to state-of-the-art (SOTA) APR methods. This work introduces a novel APR-agnostic framework, HR-APR, that formulates uncertainty estimation as cosine similarity estimation between the query and database features. It does not rely on or affect APR network architecture, which is flexible and computationally efficient. In addition, we take advantage of the uncertainty for pose refinement to enhance the performance of APR. The extensive experiments demonstrate the effectiveness of our framework, reducing 27.4% and 15.2% of computational overhead on the 7Scenes and Cambridge Landmarks datasets while maintaining the SOTA accuracy in single-image APRs.**

I. INTRODUCTION

Camera relocalisation, estimating the six degrees of freedom (6-DoF) absolute camera pose in the world space, is a core component in many applications, including mobile robotics, navigation, and augmented reality. In recent years, Absolute Pose Regressors (APRs) have emerged as an appealing approach for monocular camera pose estimation. APRs use neural networks for directly inferring 6DoF camera poses from monocular frames and offer advantages in terms of computation and memory footprint over classical 3D structure-based methods [1], [2], [3], [4], [5]. However, APR methods often struggle with generalization beyond their training data, leading to inaccurate predictions [6].

Following the first APR model, PoseNet [7], numerous methods have been proposed to enhance the robustness and accuracy of pose predictions, including modifications to network architectures [8], [9], [10], [11], [12], augmentation of the training set with labelled synthetic images [13], [10], [6], different training strategies and loss functions [14], [15], [16]. Although these methods improve accuracy, they do not distinguish inaccurate predictions. In this study, we first re-examine prevailing APRs [7], [12], [16] and demonstrate that even such state-of-the-art (SOTA) APR methods still output significantly unreliable poses, raising the need for uncertainty estimation in real-life applications.

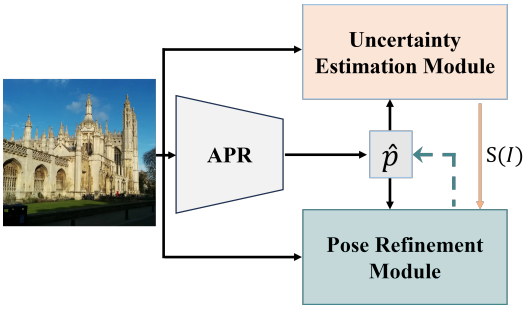


Fig. 1. HR-APR: APR-agnostic Framework with Uncertainty Estimation and Hierarchical Refinement.

Uncertainty-aware (UA) APRs aim to distinguish unreliable predictions by providing additional uncertainty information with the estimated pose. Bayesian PoseNet [17] models the uncertainty by generating multiple hypotheses for each image at inference time. In AD-PoseNet [18], uncertainty is quantified via prior guided dropout. CoordiNet [19] models heteroscedastic uncertainty during training. Deng *et al.* [20], [21] and Zangeneh *et al.* [22] use pose distributions to represent uncertainty. However, these methods can be time-consuming [17], [18], and have weak extensibility as they rely on specific network architectures and specific training schemes [21], [22], [18], [19]. Furthermore, although these UA-APRs provide both pose predictions and uncertainty estimates, the accuracy of estimate poses is much lower than SOTA non-UA APRs [12], [16] that only output poses.

Such a gap between accuracy improvement and uncertainty estimation raises the need for more modular and more generic approaches to uncertainty estimation. In this paper, we propose HR-APR, a novel test-time APR-agnostic framework with uncertainty estimation and pose refinement (see Figure 1). The uncertainty estimation module integrates a new pose-based retrieval algorithm that fetches image feature embeddings in the training set. The cosine similarity between these retrieved features and the query image is then calculated to measure uncertainty. We leverage the uncertainty to optimize an iterative pose refinement pipeline [23].

We summarize our main contributions as follows:

1) We propose a novel APR-agnostic uncertainty estimation module to predict the uncertainty of the APR output during test time. This module integrates a new pose retrieval algorithm that fetches image feature embeddings in the training set. The cosine similarity

[1] Authors are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, {cliudg,yzhaoeg,hhuangbg}@connect.ust.hk, braudt@ust.hk

[2] Tristan Braud is also with the Division of Integrated Systems Design, The Hong Kong University of Science and Technology, Hong Kong

[3] Shuai Chen and Victor Prisacariu are with the Active Vision Laboratory, Department of Engineering Science, University of Oxford, United Kingdom, {shuaic,victor}@robots.ox.ac.uk

between these retrieved features and the query image is then calculated to measure uncertainty.

2) We evaluate the accuracy of our uncertainty estimation module over three APR models with different architectures on indoor and outdoor datasets. The proposed method displays a clear correlation between pose error and uncertainty with similar performance across APR models, demonstrating the validity of the approach.
3) We further leverage the predicted uncertainty to reduce the overhead of an iterative pose refinement algorithm. HR-APR reduces the SOTA APR refinement pipeline's overhead by 27.4% and 15.2% on the indoor and outdoor datasets, respectively, while maintaining the SOTA accuracy of single-image APR methods.

II. RELATED WORK

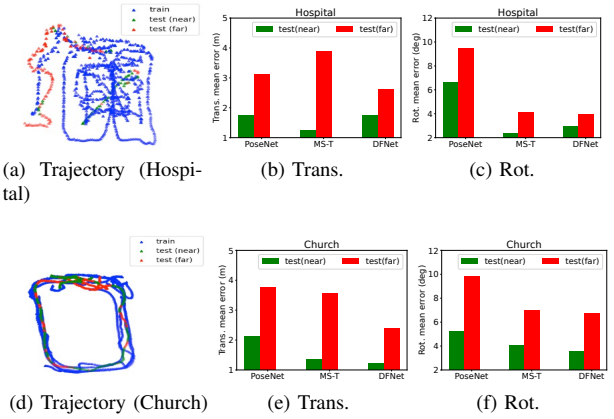
A. Absolute Pose Regression

APRs train neural networks for regressing the 6-DoF camera pose of query images. The seminal work in this area is introduced by PoseNet (PN) [7]. Further modifications to network architecures [8], [9], [10], [11], [12] and different training strategies [17], [14] have been made to improve accuracy and performance. MS-Transformer (MS-T) [12] follows MS-PN [24], which extends the single-scene APRs to multiple-scene APRs. Other approaches localise from image sequences [25], [26], [27], [28]. However, [6] has shown that most APR methods do not generalize well beyond training set via image retrieval baseline. To address this issue, [13], [10], [29] leverage additional synthetic training data. Other approaches [15], [16] adapt photometric or feature matching by applying unlabeled data with NeRF synthesis in a semi-supervised manner. However, using unlabeled data from the test set to finetune the APR network is impractical.

Most APR methods focus on improving pose prediction accuracy. However, we demonstrate in the following sections that not all pose predictions are reliable. The APRs mentioned above do not distinguish accurate predictions from poses with a large error. This paper introduces a modular, plug-and-play method to enable uncertainty estimation for non-UA APRs. Such uncertainty data can be used in the later stages of a visual positioning pipeline to improve tracking robustness and accuracy regardless of the underlying APR.

B. Uncertainty estimation

Several previous studies have investigated the pose predictions' uncertainty in the training phase of APRs. In Bayesian PoseNet [17], uncertainty is captured by quantifying the variance among multiple inferences of the same input data using Monte Carlo Dropout. Similarly, AD-PoseNet [18] evaluates pose distribution by generating multiple hypotheses via prior guided dropout. CoordiNet [19] learns heteroscedastic uncertainty as an auxiliary task during the training. Poses and uncertainties output by CoordiNet are fused into an Extended Kalman Filter (EKF) to smooth the trajectories. Deng *et al.* [20], [21] and Zangeneh *et al.* [22] use pose distributions to represent uncertainty. While these UA APRs offer both pose predictions and uncertainty estimates, the accuracy of



(a) Trajectory (Hospital) (b) Trans. (c) Rot.

(d) Trajectory (Church) (e) Trans. (f) Rot.

Fig. 2. The accuracy of APR predictions is highly corresponding to pose variants between query and training images. (a) and (d): Camera ground truth trajectory of Hospital and Church in Cambridge Landmarks dataset [7]. Blue: training set; Green: queries in the test set near the training set within 2 meters and 10 degrees; Red: queries in the test set far from the training set. (b), (c), (e), and (f) show that all three different APRs [7], [12], [16] have better predictions on test (near) than test (far).

estimate poses is much lower than SOTA non-UA APRs that only output poses, as shown in Section IV. Furthermore, existing UA APRs can be time-consuming [17], [18], require context-specific hyperparameters [21], or display weak extensibility as specific loss functions and modules need to be combined in a training scheme [21], [22], [18], [19].

Our framework enables greater flexibility regarding APR architecture and pose refinement compared to existing UA APRs. Mainstream pre-trained APRs can be integrated into our framework without modifying the network architecture or training schemes. The pose refinement module can leverage the uncertainty data to optimize computations significantly.

III. METHOD

Our objective is to enhance the robustness of APR methods and reduce the refinement overhead by identifying the reliabilities of predictions. Most existing APR methods suffer from limited generalizability due to overfitting their training sets. We first investigate several prevalent APR methods to show the accuracy of APR predictions is highly corresponding to pose variants between query and training images. The result is illustrated in Figure 2, where all three APR models exhibit more accurate predictions for test queries with viewpoints similar to those in the training set, compared to queries that fall outside the coverage of the training data. These quantitative results are also validated by the similar observation of [6], [30]. In light of this, we believe that the proximity of viewpoints also implies that these images have similar features. Therefore, we exploit pose predictions of APR to obtain the most corresponding features from the database. We then model the uncertainty of predictions by measuring the similarity between the features of the query image and the corresponding features of the database. The uncertainty information is further used as pose refinement constraints which can reduce the computational cost while achieving comparable performance.

A. Uncertainty estimation module

The uncertainty estimation module of HR-APR is shown in Figure 3. We build a database that stores low-dimensional feature embeddings of images from the training set and their associated ground truth poses, which operates as follows:

1) Given a query image I, APR P outputs estimated translation $\hat{\mathbf{x}}$ and rotation $\hat{\mathbf{q}}$ so that $P(I) = \hat{p} = <\hat{\mathbf{x}}, \hat{\mathbf{q}}>$. All feature embeddings F_r with poses within ranges of threshold d_{th} to $\hat{\mathbf{x}}$ are retrieved. Then, proceed to step 2). If there is no valid pose in the database, we simply assume that the similarity is 0 and skip to step 3).
2) A feature extractor E extracts the feature embeddings of the query image I as f_q. Then, we compute the cosine similarity between f_q and each feature embedding $f_r^i \in F_r$, retrieved in step 1). We take the maximum value as the final similarity score.
3) The estimated pose that obtain a similarity score above threshold γ is deemed to be reliable, and thus to be output directly or refined with small steps. Otherwise, the poses are considered unreliable and to be treated with more refinement steps.

Compared to typical image retrieval approaches that often require large computational complexity in the visual feature descriptor matchings, our pose retrieval is a super cheap alternative by only relying on 3-D position searches with a significant reduction in the search space. The database consisting of pose-feature embedding pairs also saves a lot of memory footprint compared to the image database in [2].

B. CNN feature embeddings

Our feature extractor E is similar to an ordinary PoseNet [7], [14], which predicts a 6-DoF camera pose for an input image. E has an FC layer before the final regressor of feature size $n = 1024$. The difference is that E also outputs the FC layer as feature embedding. E is supervised by the learnable loss function [14]:

$$\mathscr{L}_\delta = \mathscr{L}_x \exp(-\hat{s}_x) + \hat{s}_x + \mathscr{L}_q \exp(-\hat{s}_q) + \hat{s}_q \tag{1}$$

, where $\mathscr{L}_x = ||\hat{\mathbf{x}} - \mathbf{x}||_2$ and $\mathscr{L}_q = ||\frac{\hat{\mathbf{q}}}{||\hat{\mathbf{q}}||} - \mathbf{q}||_2$. To measure the difference between two feature embeddings, we compute cosine similarity between feature embedding f_q extracted from query images and each retrived feature embeddings $f_r^i, f_r^i \in F_r$.

$$\cos(f_q, f_r^i) = \frac{f_q \cdot f_r^i}{||f_q||_2 \cdot ||f_r^i||_2} \tag{2}$$

, where $f_q, f_r^i \in \mathrm{R}^{1\times 1024}$. The Similarity score of the query image I is:

$$S(I) = \max(\cos(f_q, f_r^i)), f_r^i \in F_r. \tag{3}$$

The similarity score of the query image I is $S(I)$, where $-1 \leq S(I) \leq 1$, which reflects the reliability of the prediction.

C. Outcomes

The proposed uncertainty estimation module can identify an unreliable pose $\hat{p}$ under two scenarios:

1) The training set does not contain an image close enough to the estimated location $\hat{\mathbf{x}}$.
2) Although feature embeddings of images in training set are retrieved from the database based on $\hat{\mathbf{x}}$, they all present too few similar features.

In the first scenario, $\hat{p}$ can only get 0 similarity score because of the limited generalization ability of the neural network. The APR thus has a high chance of predicting a vastly incorrect pose for an image I far from the training set. In the second scenario, although valid most-similar feature embeddings F_r extracted from images in the training set is found based on $\hat{\mathbf{x}}$, all $f_r^i \in F_r$ are not similar enough to f_q. Either the orientations of the query image and these images are very different, leading to little overlap, or the predicted $\hat{\mathbf{x}}$ and $\hat{\mathbf{q}}$ have a large error, and F_r found according to this pose is a false positive.

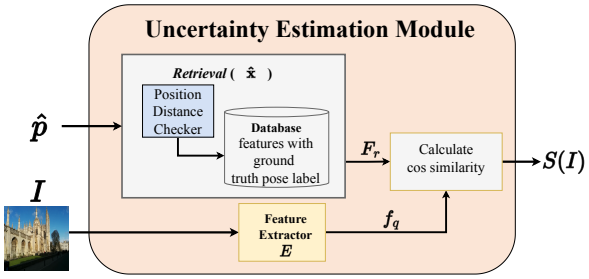


Fig. 3. Uncertainty estimation module.

D. Pose refinement module

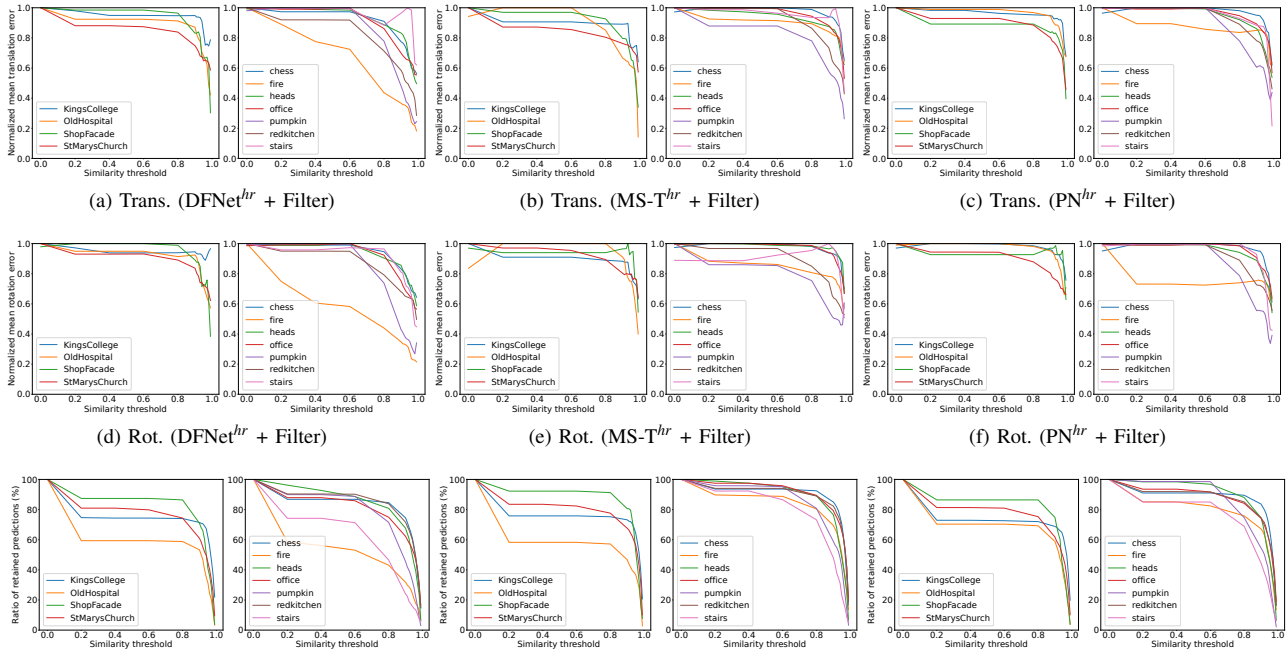
In this paper, we incorporate the test-time Neural feature synthesizer (NeFeS) refinement pipeline [23] in our pose refinement module. We modify the refinement procedure to be uncertainty-aware. The refinement procedure is as follows: (i) For estimated camera pose $\hat{p}$ with a similarity score in the uncertainty estimation module, NeFeS N renders a dense feature map m^{rend} given $\hat{p}$. (ii) At the same time, a feature map extractor G extracts a dense feature map $m^G = G(I)$ from the query image. (iii) The pose $\hat{p}$ is iteratively refined by minimizing the feature cosine similarity loss [23] between m^{rend} and m^G. The refined steps depend on the similarity score $S(I)$ in our *uncertainty estimation module*.

E. Hyperparameters

During test time, the proposed method is controlled by two hyperparameters: the distance threshold d_{th}, and the similarity threshold γ. d_{th} depends on the size of the scene.

IV. EXPERIMENT

To demonstrate the scalability and effectiveness of HR-APR, we integrate the latest representative mainstream APR architectures into our framework and test them over indoor and outdoor datasets of various scales.



(a) Trans. (DFNet^{hr} + Filter) (b) Trans. (MS-T^{hr} + Filter) (c) Trans. (PN^{hr} + Filter)

(d) Rot. (DFNet^{hr} + Filter) (e) Rot. (MS-T^{hr} + Filter) (f) Rot. (PN^{hr} + Filter)

(g) Retained Predictions (%) (DFNet^{hr} + Filter) (h) Retained Predictions (%) (MS-T^{hr} + Filter) (i) Retained Predictions (%) (PN^{hr} + Filter)

Fig. 4. Uncertainty evaluation on the 7Scenes and Cambridge Landmarks datasets. Subfigures (a)-(f) show the correlation between similarity threshold γ and normalized pose error (0-1). Based on the similarity threshold, uncertain samples with low similarity scores are gradually removed. Subfigures (g)-(i) in the last row show the ratio of retained predictions (%) as threshold increases. We observe that as we remove the samples with low similarity scores the overall error drops indicating a clear correlation between our predictions and the actual inaccurate predictions.

A. Datasets

The 7Scenes dataset [31], [32] is an indoor dataset consisting of seven small scenes from $1m^3$ to $18m^3$. Each scene contains a training set with 1000 to 7000 images and a test set with 1000 to 5000 images. We use the Structure from Motion (SfM) ground truth provided by [33] for our experiment. The Cambridge Landmarks [7] dataset represents six large-scale outdoor scenes ranging from $900m^2$ to $5500m^2$. We utilize four out of six scenes for comparative evaluation, and the SfM ground truth provided by [7]. Each scene contains 231 to 1487 images for training and 103 to 530 for testing.

B. Implementation Details

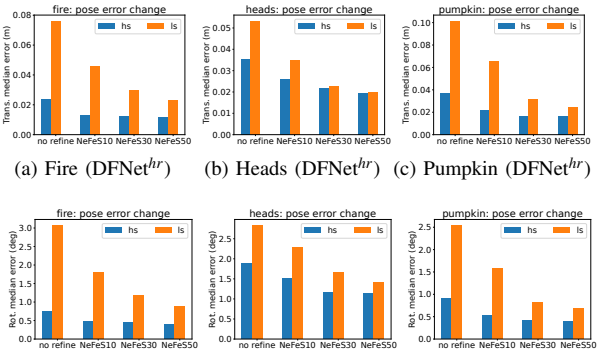
In this paper, we train the feature extractor E with EfficientNet-B0 [34] as the backbone in the *uncertainty estimation module*. The training process of E follows [7], [14]. It is trained using Adam optimizer [35] with an initial learning rate of $1e^{-4}$ and a weight decay of $5e^{-4}$. During training, all input images are resized to 256×256 and then randomly cropped to 224×224.

We implement the *uncertainty estimation module* of HR-APR over three recent APR models: **PoseNet (PN)** [7] is the classic pose regression architecture. **MS-Transformer (MS-T)** [12] aggregates the activation maps with self-attention and queries the scene-specific information. **DFNet** [16] is trained by feature matching loss. We train these three models on our platform using the open-source codes. **We incorporate these APRs into our HR-APR framework, referring to them as APR^{hr}**. We set d_{th} as $0.2m$ for the 7Scenes dataset and $1.5m$ for the Cambridge dataset. We set γ range from $0.95 \sim 0.98$. Furthermore, we show the impact of γ in the next subsection.

In this paper, we implement the NeFeS refinement pipeline [23] in our pose refinement module. The term APR^{hr} + NeFeS$^{hs(x)}_{ls(y)}$ denotes the refinement process based on the similarity of estimated poses. We refine high similarity (hs) poses that are above the similarity threshold γ with x steps. Conversely, we refine low similarity (ls) poses that are below the similarity threshold γ with y steps. For instance, APR^{hr} + NeFeS$^{hs10}_{ls50}$ indicates that we refine poses with similarity scores higher than γ for 10 steps, but refine poses with a low similarity (ls) $< \gamma$ for 50 steps. Alternatively, we use APR^{hr} + Filter to denote the scenario where we exclude the pose refinement module and directly filter out estimated poses with low similarity scores (ls). This approach helps us analyze the effectiveness of our framework, as shown in the uncertainty evaluation below. Besides, APR^{hr} + Filter can be used to reject bad predictions in real mobile robotics applications simply.

C. Uncertainty Evaluation

We compare our method quantitatively with three mainstream single-frame non-UA APR methods and UA APR methods. We utilized the resulting similarity score to measure prediction uncertainty in this paper. Our *uncertainty estimation module* applies to almost all APRs, and we present results over three different models. We gradually increase the threshold γ, filter predictions whose similarity scores are below γ, and analyze the average error of the remaining



(a) Fire (DFNet^{hr}) (b) Heads (DFNet^{hr}) (c) Pumpkin (DFNet^{hr})

(d) Fire (DFNet^{hr}) (e) Heads (DFNet^{hr}) (f) Pumpkin (DFNet^{hr})

Fig. 5. Plots of translation and rotation errors against the number of iteration for images pass the similarity threshold (hs) and images with low similarity scores (ls) do not pass the similarity threshold on some scenes of 7Scenes. The hs predictions ratio of each scene is provided in Table I. NeFeSm denotes runing the refinement process for m iterations.

TABLE I

PERCENTAGE OF PREDICTIONS WITH HIGH (0.25M, 2°), MEDIUM (0.5M, 5°), AND LOW (5M, 10°) ACCURACY [36] (HIGHER IS BETTER). THE VALUE IN PARENTHESES REPRESENTS THE RATIO OF RETAINED PREDICTIONS (HS) IDENTIFIED IN THE TEST SET BY OUR METHOD.

	Dataset	DFNet [16]	DFNet^{hr} + Filter (ours)
7Scenes	Chess	81.1/97.3/**100**	**87.6/99.6/100** (75%)
	Fire	46.5/75.7/94.5	**92.3/97.1/100** (21%)
	Heads	44.8/87.2/98.1	**56.6/98.4/100** (44%)
	Office	63.6/89.7/96.6	**82.7/97.9/99.9** (31%)
	Pumpkin	51.6/77.6/93.1	**80.5/99.1/100** (28%)
	Kitchen	59/82.1/94.5	**77.9/99.6/100** (30%)
	Stairs	32.9/79.7/98.5	**66/100/100** (14%)
Cambridge	Kings	4.1/27.1/96.5	**5.3/28.9/100** (55%)
	Hospital	**0**/1/94.5	**0/4.3/100** (13%)
	Shop	7.8/31.1/97.1	**19/50/100** (16%)
	Church	0.9/9.1/87.9	**1.9/14.6/99.1** (40%)

sample, as shown in Figure 4. The subfigures Figure 4.g-i indicate a decrease in the ratio of retained predictions as the threshold γ increases, coupled with the decrease in normalized mean translation and rotation errors of the retained predictions as γ increases in Figures 4.a-f. This clear correlation demonstrates that samples with higher similarity scores tend to yield more accurate estimations.

1) 7Scenes: For all three APR models, when the similarity threshold γ is higher than 0.95, the translation and rotation mean errors are reduced by 25% to 80% across the scenes.

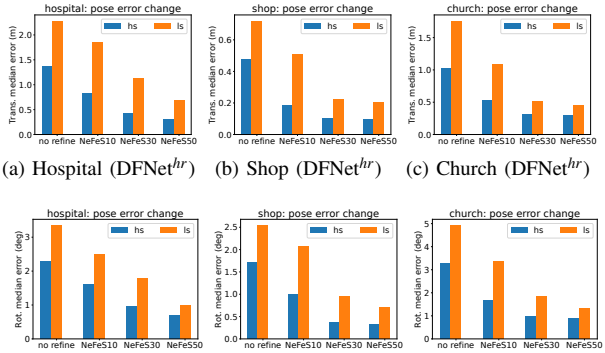
2) Cambridge: For all three APR models, when γ is higher than 0.95, the translation and rotation mean errors are reduced by 15% to 80% on different scenes.

Non-UA APRs (MS-T and DFNet) exhibit higher accuracy individually compared to UA APRs (Bayesian PN, AD-PN, BMDN, and VaPoR) as shown in Table II and Table III. Our new APR-agnostic *uncertainty estimation module* enables uncertainty estimation for such models, allowing us to leverage their greater accuracy in applications that require uncertainty measures.

D. UA pose refinement evaluation

Adding uncertainty awareness into APRs through the *uncertainty estimation module* enables filtering unreliable poses and prioritizing the optimization of less accurate predictions.

1) 7Scenes: Table I illustrates the performance improvement provided by solely filtering unreliable poses identified by the pose estimation module. DFNet^{hr}+Filter achieves higher percentages in all accuracy levels by keeping the $14 \sim 75\%$ predictions with reasonably high similarity scores pass γ. Furthermore, we completely filter out the pose predictions in the low accuracy level $(5m, 10°)$ in six of seven scenes, and reduce it to 0.1% in the Office scene. Regarding pose optimization, Figure 5 shows that high similarity (hs) predictions, with similarity scores above γ, largely converge within 10 steps of optimization using the NeFeS method. On the other hand, low similarity (ls) predictions still require 10 to 50 steps to converge.



(a) Hospital (DFNet^{hr}) (b) Shop (DFNet^{hr}) (c) Church (DFNet^{hr})

(d) Hospital (DFNet^{hr}) (e) Shop (DFNet^{hr}) (f) Church (DFNet^{hr})

Fig. 6. Plots of translation and rotation errors against the number of iteration for images pass the similarity threshold (hs) and images with low similarity scores (ls) do not pass the similarity threshold on some scenes of Cambridge. The hs predictions ratio of each scene is provided in Table I. NeFeSm denotes runing the refinement process for m iterations.

2) Cambridge: Similar to 7Scenes, Table I shows that filtering unreliable poses achieves higher percentages in all accuracy levels by keeping the $13 \sim 55\%$ predictions with reasonably high similarity scores pass γ. Besides, we filter out the pose predictions in low accuracy level $(5m, 10°)$ in three of four scenes and reduce it to 0.9% in the Church. As shown in Figure 6, high similarity (hs) predictions, with similarity scores above γ, largely converge within 30 steps of optimization using the NeFeS method, but low similarity (ls) predictions still require 30 to 50 steps to converge.

Table II and Table III show the accuracy of our UA pose refinement method compared to the SOTA non-UA APRs and refinement pipeline. The results demonstrate that our method achieves comparable accuracy to SOTA methods while reducing the optimization overhead by 27.4% and 15.2% on the indoor and outdoor datasets, respectively.

E. Analysis

Figure 2 shows that all three APR models have higher accuracy on query images closer to the training set's viewpoints. Without knowing the ground truth of the query image,

TABLE II

COMPARISONS ON 7-SCENES DATASET. THE MEDIAN TRANSLATION AND ROTATION ERRORS (M/°) OF DIFFERENT METHODS AND THE AVERAGE REFINE STEPS FOR EACH QUERY IMAGE. THE BEST RESULTS ARE IN BOLD (LOWER IS BETTER).

	Methods	Chess	Fire	Heads	Office	Pumpkin	Redkitchen	Stairs	Avg. [$m/°$]	Avg. Steps
UA APR	Bayesian PN [17]	0.37/7.24	0.43/13.7	0.31/12.0	0.48/8.04	0.61/7.08	0.58/7.54	0.48/13.1	0.47/9.81	-
	CoordiNet [19]	0.14/6.7	0.27/11.6	0.13/13.6	0.21/8.6	0.25/7.2	0.26/7.5	0.28/12.9	0.22/9.70	-
	BMDN[1][21]	0.10/6.47	0.26/14.8	0.13/13.4	0.19/9.73	0.20/9.40	0.19/10.9	0.34/14.1	0.20/11.26	-
	VaPoR [22]	0.17/6.9	0.30/14.1	0.17/14.5	0.24/9.3	0.30/8.3	0.26/10.2	0.47/15.5	0.27/11.26	-
non-UA APR	PN [7]	0.10/4.02	0.28/9.44	0.18/13.9	0.17/5.99	0.22/5.18	0.23/5.91	0.34/11.6	0.22/7.16	-
	MS-T [12]	0.11/6.45	0.23/10.9	0.13/13.1	0.18/8.18	0.16/6.87	0.17/8.44	0.30/10.4	0.18/9.19	-
	DFNet [16]	0.03/1.12	0.06/2.30	0.04/2.29	0.06/1.54	0.07/1.92	0.07/1.74	0.12/2.63	0.06/1.93	-
APR Refine	DFNet + NeFeS10 [23]	**0.02/0.55**	0.03/1.20	0.03/1.88	0.04/1.08	0.04/0.98	0.03/0.97	0.10/2.47	0.04/1.30	10
	DFNet + NeFeS50 [23]	**0.02**/0.57	**0.02/0.74**	**0.02/1.28**	**0.02/0.56**	**0.02/0.55**	**0.02/0.57**	**0.05/1.28**	**0.02/0.79**	50
Ours	DFNet^{hr} + NeFeS$^{10}_{50}$	**0.02/0.55**	**0.02**/0.75	**0.02**/1.45	**0.02**/0.64	**0.02**/0.62	**0.02**/0.67	**0.05**/1.30	**0.02**/0.85	36.3

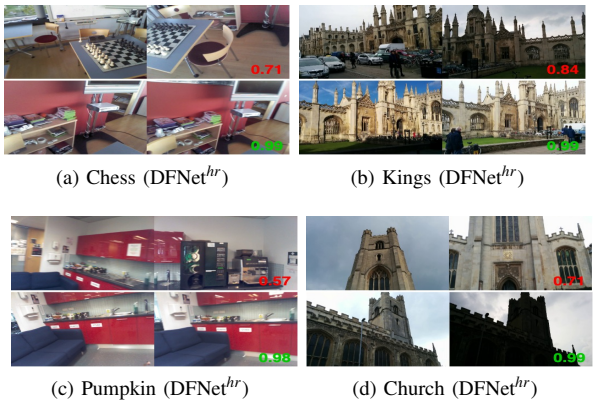
[1] Results of BMDN taken from [22].

TABLE III

COMPARISONS ON CAMBRIDGE DATASET. THE MEDIAN TRANSLATION AND ROTATION ERRORS (M/°) OF DIFFERENT METHODS AND THE AVERAGE REFINE STEPS FOR EACH QUERY IMAGE. BEST RESULTS ARE IN BOLD (LOWER IS BETTER).

	Methods	Kings	Hospital	Shop	Church	Avg.	Avg. Steps
UA APR	Bayesian PN	1.74/ 4.06	2.57/ 5.14	1.25/ 7.54	2.11/ 8.38	1.92/6.28	-
	AD-PN	1.3/1.67	2.28/4.80	1.22/6.17	-/-	-/-	-
	BMDN[1]	1.51/2.14	2.25/3.93	3.52/5.41	2.16/5.99	2.36/4.37	-
	VaPoR	1.65/2.88	2.06/4.33	1.02/6.03	1.80/5.90	1.63/4.79	-
non-UA APR	PN	0.93/2.73	2.24/7.88	1.47/6.62	2.37/5.94	1.75/5.79	-
	MS-T	0.85/1.45	1.75/2.43	0.88/3.20	1.66/4.12	1.29/2.80	-
	DFNet	0.73/2.37	2.00/2.98	0.67/2.21	1.37/4.02	1.19/2.90	-
APR Refine	DFNet + NeFeS30	0.37/0.64	0.98/1.61	0.17/0.60	0.42/1.38	0.49/1.06	30
	DFNet + NeFeS50	0.37/**0.54**	**0.52/0.88**	0.15/0.53	**0.37/1.14**	**0.35/0.77**	50
Ours	DFNet^{hr} + NeFeS$^{30}_{50}$	**0.36**/0.58	0.53/0.89	**0.13/0.51**	0.38/1.16	**0.35**/0.78	42.4

[1] Results of BMDN taken from [22].



(a) Chess (DFNet^{hr}) (b) Kings (DFNet^{hr})

(c) Pumpkin (DFNet^{hr}) (d) Church (DFNet^{hr})

Fig. 7. Examples of retrieved image pairs in Cambridge and 7Scenes. For each pair, left side is the query image in test set, right side is the training set image retrieved by pose-based algorithm with the maximum similarity. The red numbers in the upper row of each subfigure represent the two images with low similarity and the green numbers in the lower row represent the two images with high similarity.

our uncertainty estimation module utilizes image features to measure the similarity between the query image and the training set, as we expect the similarity of image features to equate the proximity of viewpoints. We visualize the query image and the corresponding image in the training set retrieved by our pose-based algorithm with the maximum similarity. As shown in Figure 7, some query images are very similar to the training set images, while others are less straightforward to match. The clear correlation in Figure 4 and improvement in all accuracy levels after filtering in Table I confirm our uncertainty estimation module is effective and fits the design outcomes in Section III-C. Besides, Figure 7.b and Figure 7.d show that the high-level and low-dimensional feature embeddings extracted from PoseNet-based feature extractor E are robust to different lighting, weather, moving objects, and pedestrians where point-based SIFT registration fails [7].

F. System Efficiency

We evaluate the processing time of the proposed framework on a PC equipped with an NVIDIA GeForce GTX 3090 GPU. Based on the implementation in Section III and Section IV, we repeat each measurement 1000 times. The feature extraction of E takes 3.4ms. Pose-based feature retrieval with $\mathcal{O}(n)$ complexity takes 1.98ms, while similarity calculation takes 1.3ms in the Heads dataset. Overall, the *uncertainty estimation module* only adds about an extra 6.7ms to each inference time of APRs. That means predictions can get the uncertainty in real-time. The feature extractor E with EfficientNetb0 as backbone only takes 20 MB to store the weights, and each 1×1024 feature embedding of an image only takes 4.2 KB. That means our *uncertainty estimation module* is also very storage efficient. Through its high modularity, our framework provides a new paradigm for real-life camera relocalisation with APRs. It can integrate most existing and future APRs and test-time refinement methods with minimal changes.

V. CONCLUSIONS

We introduce an APR-agnostic framework, HR-APR, which includes an uncertainty estimation module that uses a novel pose retrieval algorithm to calculate the cosine similarity between image feature embeddings in the training set and the query image. This uncertainty estimation is used to optimize an iterative pose refinement algorithm. We evaluate the performance of our method on indoor and outdoor datasets using three different APRs. Our results demonstrate a clear correlation between pose error and uncertainty, validating the effectiveness of our approach. Moreover, HR-APR significantly reduces the computational overhead of the refinement pipeline while maintaining accuracy.

REFERENCES

[1] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-net: A trainable cnn for joint description and detection of local features," in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2019, pp. 8092–8101.

[2] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 716–12 725.

[3] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii, "Inloc: Indoor visual localization with dense matching and view synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7199–7209.

[4] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3456–3465.

[5] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient & effective prioritized matching for large-scale image-based localization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 9, pp. 1744–1756, 2016.

[6] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe, "Understanding the limitations of cnn-based absolute camera pose regression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3302–3312.

[7] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2938–2946.

[8] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using lstms for structured feature correlation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 627–637.

[9] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Image-based localization using hourglass networks," in *Proceedings of the IEEE international conference on computer vision workshops*, 2017, pp. 879–886.

[10] J. Wu, L. Ma, and X. Hu, "Delving deeper into convolutional neural networks for camera relocalization," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5644–5651.

[11] B. Wang, C. Chen, C. X. Lu, P. Zhao, N. Trigoni, and A. Markham, "Atloc: Attention guided camera localization," *arXiv preprint arXiv:1909.03557*, 2019.

[12] Y. Shavit, R. Ferens, and Y. Keller, "Learning multi-scene absolute pose regression with transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2733–2742.

[13] T. Naseer and W. Burgard, "Deep regression for monocular camera-based 6-dof global localization in outdoor environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1525–1530.

[14] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5974–5983.

[15] S. Chen, Z. Wang, and V. Prisacariu, "Direct-posenet: absolute pose regression with photometric consistency," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 1175–1185.

[16] S. Chen, X. Li, Z. Wang, and V. A. Prisacariu, "Dfnet: Enhance absolute pose regression with direct feature matching," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*. Springer, 2022, pp. 1–17.

[17] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *2016 IEEE international conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4762–4769.

[18] Z. Huang, Y. Xu, J. Shi, X. Zhou, H. Bao, and G. Zhang, "Prior guided dropout for robust visual localization in dynamic environments," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2791–2800.

[19] A. Moreau, N. Piasco, D. Tsishkou, B. Stanciulescu, and A. de La Fortelle, "Coordinet: uncertainty-aware pose regressor for reliable vehicle localization," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2229–2238.

[20] H. Deng, M. Bui, N. Navab, L. Guibas, S. Ilic, and T. Birdal, "Deep bingham networks: Dealing with uncertainty and ambiguity in pose estimation," *International Journal of Computer Vision*, vol. 130, no. 7, pp. 1627–1654, 2022.

[21] M. Bui, T. Birdal, H. Deng, S. Albarqouni, L. Guibas, S. Ilic, and N. Navab, "6d camera relocalization in ambiguous scenes via continuous multimodal inference," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer, 2020, pp. 139–157.

[22] F. Zangeneh, L. Bruns, A. Dekel, A. Pieropan, and P. Jensfelt, "A probabilistic framework for visual localization in ambiguous scenes," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3969–3975.

[23] S. Chen, Y. Bhalgat, X. Li, J. Bian, K. Li, Z. Wang, and V. A. Prisacariu, "Refinement for absolute pose regression with neural feature synthesis," *arXiv preprint arXiv:2303.10087*, 2023.

[24] H. Blanton, C. Greenwell, S. Workman, and N. Jacobs, "Extending absolute pose regression to multiple scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 38–39.

[25] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Geometry-aware learning of maps for camera localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2616–2625.

[26] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen, "Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6856–6864.

[27] N. Radwan, A. Valada, and W. Burgard, "Vlocnet++: Deep multitask learning for semantic visual localization and odometry," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4407–4414, 2018.

[28] A. Valada, N. Radwan, and W. Burgard, "Deep auxiliary learning for visual localization and odometry," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 6939–6946.

[29] A. Moreau, N. Piasco, D. Tsishkou, B. Stanciulescu, and A. de La Fortelle, "Lens: Localization enhanced by nerf synthesis," in *Conference on Robot Learning*. PMLR, 2022, pp. 1347–1356.

[30] T. Ng, A. Lopez-Rodriguez, V. Balntas, and K. Mikolajczyk, "Reassessing the limitations of cnn methods for camera pose regression," *arXiv preprint arXiv:2108.07260*, 2021.

[31] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi, "Real-time rgb-d camera relocalization," in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2013, pp. 173–179.

[32] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in rgb-d images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2930–2937.

[33] E. Brachmann, M. Humenberger, C. Rother, and T. Sattler, "On the limits of pseudo ground truth in visual camera re-localisation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6218–6228.

[34] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[36] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, *et al.*, "Benchmarking 6dof outdoor visual localization in changing conditions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8601–8610.

Statement of Authorship for the paper "HR-APR: APR-agnostic Framework with Uncertainty Estimation and Hierarchical Refinement for Camera Relocalisation" in Chapter 6.

Paper title	HR-APR: APR-agnostic Framework with Uncertainty Estimation and Hierarchical Refinement for Camera Relocalisation
Authors	Changkun Liu, **Shuai Chen**, Yukun Zhao, Huajian Huang, Victor Prisacariu, Tristan Braud, Victor Adrian Prisacariu
Publication status	Published
Publication details	International Conference on Robotics and Automation (ICRA), 2024.

Student Confirmation

Student name	Shuai Chen	
Contribution to the paper	Second-author contribution: • Conception of research ideas • Design of models • Running of some experiments • Writing and presentation of the paper	
Signature and Date		Apr. 23th 2025

Supervisor Confirmation

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

Supervisor name	Prof. Victor Adrian Prisacariu	
Supervisor comments	The description is accurate	
Signature and Date		Apr. 23th 2025

Chapter 7

Efficient Camera Pose Refinement via 3D Gaussian Splatting and 3D Foundation Model

This chapter introduces a more efficient and robust method for camera pose refinement. In the previous two chapters, we show that NeRF-based refinement methods can effectively improve coarse pose estimates produced by APR. Conceptually, such methods share similarities with traditional image-retrieval-based pipelines, such as HLoc, which rely on place recognition for coarse localisation. In contrast to traditional methods constrained by the discrete poses of the retrieved database images, NeRF-based refinement takes advantage of the continuous and more accurate initial poses provided by APR. However, these methods often fail to improve the already accurate poses predicted by geometry-based methods such as SCR [92, 174]. When the initial estimate is close to ground truth, NeRF-based iterative refinement methods sometimes become unreliable. One root cause to this phenomenon is that the optimisation signal becomes weak, noisy, and highly sensitive to small errors due to the artefacts produced by the view synthesis process,

limiting the refinement precision.

Additionally, methods like NeFeS [29] and CrossFire [113] suffer from high computational overhead due to slow iterative rendering and backpropagation at the test time. They also require training scene-specific feature descriptors, which is tedious and time-consuming. These challenges highlight the need for a more general, efficient, and scalable camera pose refinement method.

To address this, we present a more effective pose refinement framework based on our recent paper published at the *2025 International Conference on Learning Representations*, titled *GS-CPR: Efficient Camera Pose Refinement via 3D Gaussian Splatting*. GS-CPR introduces several upgrades compared to the previous NeFeS project. First, we replace the NeRF model with a 3D Gaussian Splatting model, enabling efficient rendering of high-quality synthetic images and depth maps. Second, it avoids iterative feature-metric refinement by establishing dense 2D-3D correspondences, enabling one-shot pose refinement without iterative optimisation. Third, it eliminates the tedious feature descriptor training process by using a 3D foundation model MASt3R [88] for direct 2D-2D matching.

Together, these design choices make GS-CPR faster, more generalisable, and easier to deploy. Our experiments show that GS-CPR consistently achieves state-of-the-art results on both indoor and outdoor benchmarks. It not only improves pose estimates from APR methods, but also enhances the accuracy of state-of-the-art SCR methods such as ACE [92] and GLACE [174]. On average, GS-CPR reduces pose estimation errors by up to 30% and runs in less than 0.2 seconds per query, offering more than 50x speed up compared to NeFeS in Chapter 5.

7.1 Discussion on Methodology and Constraints

To provide a more comprehensive understanding of the proposed GS-CPR and the preceding refinement frameworks, this section clarifies the requirements for initial poses, label types, and the inherent constraints regarding scene scalability.

Initial Pose and Convergence. The efficacy of NeRF and 3DGS-based refinement is fundamentally contingent upon the quality of the initial pose estimates. As demonstrated in the convergence analyses in Chapters 4, 5 and 7, these methods typically require an initial APR estimate to fall within a specific convergence radius to avoid local minima. Notably, as the underlying scene representation evolved from NeRF to 3D-GS in this chapter, we observed an increased tolerance threshold, enabling the system to successfully refine even coarser initial estimates due to the higher-fidelity geometric signals provided by the latest 3D-GS solution.

Label Requirements and Scene Scale. Our frameworks are compatible with various pose labels, including those generated via SLAM or SfM, although SfM-based labels generally yield higher refinement precision due to their superior global quality. Regarding scalability, while our methods demonstrate robust performance on standard benchmarks, we acknowledge that the datasets used consist primarily of small-to-medium scale scenes. As the scene volume expands, there is a commensurate increase in the required model capacity (e.g., the number of Gaussians or parameters) and label density to maintain high-frequency details.

Limitations and Robustness. Despite recent advances, representations such as NeRF and 3DGS still face challenges in extreme scenarios, including city-scale environments, day-to-night transitions, and seasonal variations. Consequently, our refinement methods are most effective within scenes that the underlying representation can accurately reconstruct. We anticipate that these constraints will be mitigated as the research community continues to improve the quality and effi-

ciency of 3D scene representations. Further discussions on these scalability challenges and potential multi-modal solutions are provided in the future work section in Chapter 9.

GS-CPR: EFFICIENT CAMERA POSE REFINEMENT VIA 3D GAUSSIAN SPLATTING

Changkun Liu[1]* **Shuai Chen**[2] **Yash Bhalgat**[2] **Siyan Hu**[1] **Ming Cheng**[3]
Zirui Wang[2] **Victor Adrian Prisacariu**[2] **Tristan Braud**[1]
[1]HKUST [2]University of Oxford [3]Dartmouth College

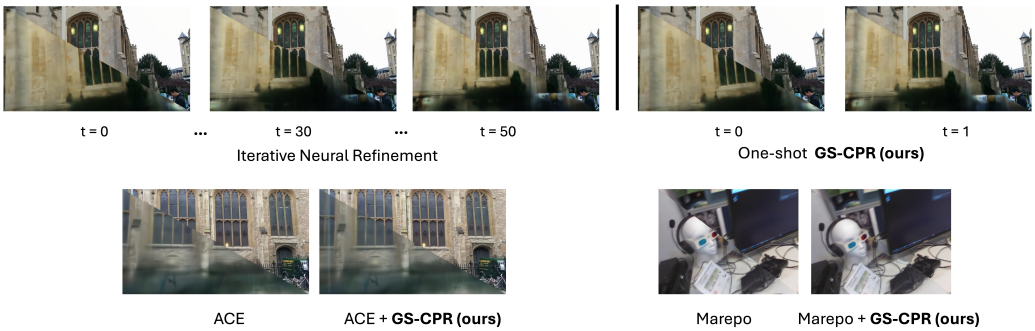


Figure 1: GS-CPR refines pose predictions of state-of-the-art APR and SCR models in a one-shot manner, achieving greater accuracy compared to the iterative neural refinement method, such as NeFeS (Chen et al., 2024a). Each subfigure is divided by a diagonal line, with the **bottom left** part rendered using the estimated/refined pose and the **top right** part displaying the ground truth image.

ABSTRACT

We leverage 3D Gaussian Splatting (3DGS) as a scene representation and propose a novel test-time camera pose refinement (CPR) framework, GS-CPR. This framework enhances the localization accuracy of state-of-the-art absolute pose regression and scene coordinate regression methods. The 3DGS model renders high-quality synthetic images and depth maps to facilitate the establishment of 2D-3D correspondences. GS-CPR obviates the need for training feature extractors or descriptors by operating directly on RGB images, utilizing the 3D foundation model, MASt3R, for precise 2D matching. To improve the robustness of our model in challenging outdoor environments, we incorporate an exposure-adaptive module within the 3DGS framework. Consequently, GS-CPR enables efficient one-shot pose refinement given a single RGB query and a coarse initial pose estimation. Our proposed approach surpasses leading NeRF-based optimization methods in both accuracy and runtime across indoor and outdoor visual localization benchmarks, achieving new state-of-the-art accuracy on two indoor datasets. The project page is available at: `https://xrim-lab.github.io/GS-CPR/`.

1 INTRODUCTION

Camera relocalization, the task of determining the 6-DoF camera pose within a given environment based on a query image, is critical for numerous applications, including robotics, autonomous vehicles, augmented reality, and virtual reality. Current methods for camera pose estimation primarily fall into the categories of structure-based approaches and absolute pose regression (APR) techniques. Classic structure-based pipelines (Dusmanu et al., 2019; Sarlin et al., 2019; Taira et al., 2018; Noh et al., 2017; Sattler et al., 2016; Sarlin et al., 2020; Lindenberger et al., 2023) rely on 2D-3D correspondences between a point cloud and the reference image. Another class of structure-based

*cliudg@connect.ust.hk, research conducted during a visit at Active Vision Lab, University of Oxford.

methods - Scene Coordinate Regression (SCR) (Brachmann et al., 2017; 2023; Wang et al., 2024a; Brachmann & Rother, 2021) - uses neural networks for direct regression of 2D-3D correspondences. These 2D-3D correspondences are fed into Perspective-n-Point (PnP) (Gao et al., 2003) for pose estimation. APR methods (Kendall et al., 2015; Wang et al., 2019; Chen et al., 2021; Shavit et al., 2021) employ neural networks to infer camera poses from query images directly. While APR approaches offer fast inference times, they often struggle with accuracy and generalization (Sattler et al., 2019; Liu et al., 2024a). SCR methods generally achieve higher accuracy but at the cost of increased computational complexity.

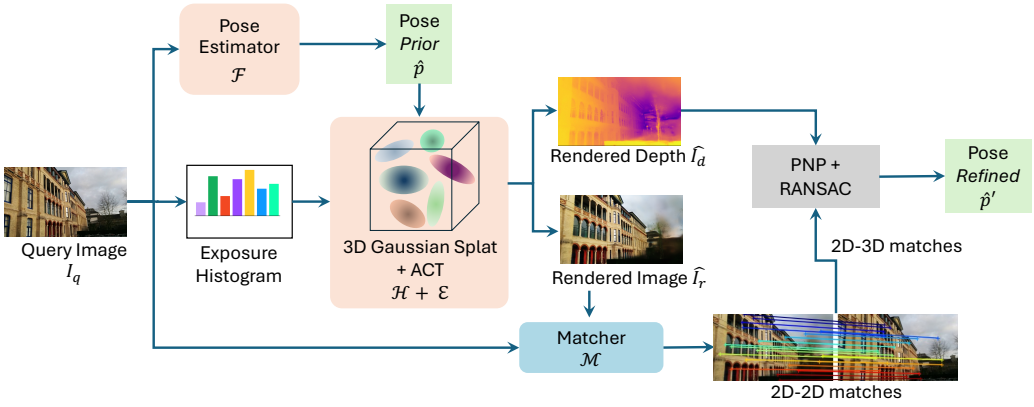


Figure 2: Overview of GS-CPR. We assume the availability of a pre-trained pose estimator $\mathcal{F}$ and a pre-trained 3DGS model $\mathcal{H}$ of the scene. For a query image I_q, we first obtain an initial estimated pose $\hat{p}$ from the pose estimator $\mathcal{F}$. Our goal is to output a refined pose $\hat{p}'$.

Given the above limitations, there has been a growing interest in pose refinement methods to enhance the accuracy of the *initial* pose estimates of an underlying pose-estimation method. Recent approaches have leveraged Neural Radiance Fields (NeRF) (Mildenhall et al., 2020) for this purpose. For instance, NeFeS (Chen et al., 2024a) proposes a test-time refinement pipeline. However, it offers limited improvements in accuracy and suffers from slow convergence due to the computational demands of NeRF rendering and the requirement for backpropagation through the pose estimation model. Furthermore, a recent NeRF-based localization method - CrossFire (Moreau et al., 2023) - establishes explicit 2D-3D matches using features rendered from NeRF. However, training a customized scene model together with the scene-specific localization descriptor is required, and it exhibits lower accuracy compared to classic structure-based methods.

To address the challenges of slow convergence, limited accuracy, and the need for training customized feature descriptors, we propose a novel test-time pose refinement framework, termed GS-CPR, as illustrated in Figure 1 and Figure 2. GS-CPR employs 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) for scene representation and leverages its high-quality, fast novel view synthesis (NVS) capabilities to render images and depth maps. This facilitates the efficient establishment of 2D-3D correspondences between the query image and the rendered image, based on the initial pose estimate from the underlying pose estimator (e.g., APR, SCR). We incorporate an exposure-adaptive module into the 3DGS model to improve its robustness to the domain shift between the query image and the rendered image. Secondly, our method operates directly on RGB images, utilizing the 3D vision foundation model MASt3R (Leroy et al., 2024) for precise matching, eliminating the need for training scene-specific feature extractors or descriptors (Chen et al., 2024a; Moreau et al., 2023). This significantly accelerates our method compared to iterative NeRF-based refinement methods (Chen et al., 2024a) and makes our framework easier to deploy than CrossFire (Moreau et al., 2023) and its variants (Zhou et al., 2024; Liu et al., 2023; Zhao et al., 2024).

Lastly, we conduct comprehensive quantitative evaluations and ablation studies on the 7Scenes (Glocker et al., 2013; Shotton et al., 2013), 12Scenes (Valentin et al., 2016), and Cambridge Landmarks (Kendall et al., 2015) benchmarks. GS-CPR significantly enhances the pose estimation accuracy of both APR and SCR methods across these benchmarks, achieving new state-of-the-art accuracy on the two indoor datasets. Unlike previous NeRF-based methods (Chen et al., 2024a),

which fail to improve SCR methods, such as ACE (Brachmann et al., 2023), our method offers substantial improvements and outperforms other leading NeRF-based methods (Germain et al., 2022; Moreau et al., 2023; Zhou et al., 2024; Liu et al., 2023; Zhao et al., 2024).

2 Related Work

Pose Estimation without 3D Representation. A straightforward approach for coarse pose estimation is using image retrieval (Arandjelovic et al., 2016; Ge et al., 2020; Gordo et al., 2017) to average poses from top-retrieved images, but this lacks precision. Absolute Pose Regression (APR) methods (Kendall et al., 2015; Kendall & Cipolla, 2016; 2017; Wang et al., 2019; Chen et al., 2021; 2022; Shavit et al., 2021; Chen et al., 2024b; Lin et al., 2024) directly regress a pose from a query image using trained models, bypassing 3D representations and geometric relationships. Despite being fast, APR methods suffer in accuracy and generalization (Sattler et al., 2019; Liu et al., 2024a) compared to structure-based techniques. LENS (Moreau et al., 2022) enhances APR by augmenting views with NeRF, but matching the accuracy of 3D structure-based methods remains challenging. To improve APR methods' accuracy, we used 3DGS as a 3D representation and utilized its geometry information to optimize the initial prediction.

Structure-based Pose Estimation. Classical 3D structure-based methods, like the hierarchical localization pipeline (HLoc) (Dusmanu et al., 2019; Sarlin et al., 2019; Taira et al., 2018; Noh et al., 2017; Sattler et al., 2016; Sarlin et al., 2020; Lindenberger et al., 2023), predict camera poses using a point cloud and a database of reference images, requiring descriptor storage and 2D-3D correspondence through image retrieval. In contrast, Scene Coordinate Regression (SCR) methods (Brachmann et al., 2017; 2023; Wang et al., 2024a; Brachmann & Rother, 2021) directly regress 2D-3D correspondences using neural networks and apply PnP (Gao et al., 2003) and RANSAC (Fischler & Bolles, 1981) for pose estimation. Our GS-CPR eliminates the need for reference images and descriptor databases by using a 3DGS model for scene representation, further optimizing SCR outputs like ACE (Brachmann et al., 2023).

NeRF-based Pose Estimation. NeRF-based pose estimation methods (Chen et al., 2024a; Yen-Chen et al., 2021; Lin et al., 2023) rely on iterative rendering and pose updates, leading to slow convergence and limited accuracy. While NeFeS (Chen et al., 2024a) improves APR pose estimation, it faces difficulties in enhancing SCR results and suffers from long refinement runtime. HR-APR (Liu et al., 2024a) speeds up optimization by 30%, but the average runtime of each query still takes several seconds on a high-performance GPU. Other NeRF-based methods like FQN (Germain et al., 2022), CrossFire (Moreau et al., 2023), NeRFLoc (Liu et al., 2023), and NeRFMatch (Zhou et al., 2024) improve positioning by establishing 2D-3D matches but require specialized feature extractors and suffer from slow rendering and quality issues.

3DGS-based Pose Estimation. With the NVS field transitioning from NeRF to 3DGS, methods proposed by Sun et al. (2023) and Botashev et al. (2024) refine camera poses in an inefficient iterative manner by inverting 3DGS, following iNeRF (Yen-Chen et al., 2021). In contrast, 6DGS (Bortolon et al., 2024) achieves a one-shot estimate by projecting rays from an ellipsoid surface, avoiding iteration. Although both methods use 3DGS for visual localization, neither has been tested on large benchmarks (Kendall et al., 2015; Valentin et al., 2016) or compared with mainstream methods like SCR and APR. We propose an approach using 3DGS for 2D-3D correspondences, similar to CrossFire (Moreau et al., 2023), but without requiring training feature extractors or feature matchers. Our method generates high-quality synthetic images and employs direct 2D-2D matching, making it faster and easier to deploy than previous NeRF-based methods such as NeFeS, CrossFire, and other variants (Germain et al., 2022; Zhou et al., 2024; Liu et al., 2023; 2024a; Zhao et al., 2024).

3 Proposed Method

GS-CPR is a test-time camera pose refinement framework. We assume the availability of a pretrained pose estimator and a 3DGS model of the scene. For a query image, we first obtain an initial estimated pose from the pose estimator. Our goal is to output a refined pose.

Given a query image $I_q \in \mathbb{R}^{H \times W \times 3}$ with camera intrinsics $K \in \mathbb{R}^{3 \times 3}$, a pose estimator $\mathcal{F}$ (typically an APR or SCR model) predicts an *initial* 6-DoF pose $\hat{p} = [\hat{\mathbf{t}}|\hat{\mathbf{R}}]$, where $\hat{\mathbf{t}} \in \mathbb{R}^3$ and $\hat{\mathbf{R}} \in \mathbb{R}^{3 \times 3}$

represent the estimated translation and rotation respectively. Subsequently, for the viewpoint $\hat{p}$, a pretrained 3DGS model $\mathcal{H}$ renders an image $\hat{I}_r \in \mathbb{R}^{H\times W\times 3}$ and a depth map $\hat{I}_d \in \mathbb{R}^{H\times W\times 1}$. We use an exposure-adaptive affine color transformation (ACT) module $\mathcal{E}$ during this rendering process to enhance the robustness of our model to challenging outdoor environments (see Section 3.1). A matcher $\mathcal{M}$ then establishes dense 2D-2D correspondences between I_q and $\hat{I}_r$. Then we can establish the 2D-3D matches based on $\hat{I}_q$ and $\hat{I}_d$ (see Section 3.2). Finally, we obtain the refined pose $\hat{p}'$ from these 2D-3D matches (see Section 3.2). An overview of our framework is depicted in Figure 2. We also explore a faster pose refinement framework without 2D-3D matches depicted in Figure 3 (see Section 3.3).

3.1 3DGS Test-time Exposure Adaptation

Existing literature (Kerbl et al., 2023; Lu et al., 2024) shows that 3DGS achieves high-quality novel view renderings but assumes training and testing without significant photometric distortions. In visual relocalization, mapping and query sequences often differ in lighting due to varying times, weather, and exposure. This creates a significant appearance gap between 3DGS renderings and query images, negatively impacting 2D-2D matching performance.

To address this issue, we apply an exposure-adaptive affine color transformation module $\mathcal{E}$ (Chen et al., 2022; 2024a) to 3DGS, allowing the 3DGS to adaptively render appearances during testing and accurately reflect the exposure of I_q. Specifically, we use a 4-layer MLP that takes the luminance histogram of the query image as input and produces a 3x3 matrix $\mathbf{Q}$ along with a 3-dimensional bias vector $\mathbf{b}$. These outputs are then directly applied to the rendered pixels of the 3DGS as shown in Equation 1, ensuring a closer match to the exposure of the query image.

$$\hat{\mathbf{C}}(\mathbf{r}) = \mathbf{Q}\hat{\mathbf{C}}_{\text{rend}}(\mathbf{r}) + \mathbf{b}, \tag{1}$$

where $\hat{\mathbf{C}}(\mathbf{r})$ is the final per-pixel color and $\hat{\mathbf{C}}_{\text{rend}}(\mathbf{r})$ is the rendered per-pixel color obtained from the 3DGS model $\mathcal{H}$.

3.2 Pose Refinement with 2D-3D Correspondences

GS-CPR estimates the camera pose by establishing 2D-3D correspondences between the query image I_q and the scene representation. This process involves the following steps:

2D-2D Matching. First, an image $\hat{I}_r$ is rendered from the initial estimated viewpoint $\hat{p}$. A Matcher $\mathcal{M}$ is then used to establish 2D-2D pixel correspondences $C_{q,r}$ between the query image I_q and the rendered image $\hat{I}_r$. In our implementation, the matcher $\mathcal{M}$ is a recently released 3D vision foundation model, MASt3R (Leroy et al., 2024). MASt3R demonstrates strong robustness for 2D-2D matching across image pairs with the sim-to-real domain gap.

3D Coordinate Map Generation. Simultaneously, we use our trained 3DGS model $\mathcal{H}$ to render a depth map $\hat{I}_d$ from the viewpoint $\hat{p}$. We modify the rasterization engine of 3DGS to render the depth map as follows:

$$\hat{I}_d = \sum_{i\in N} d_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \tag{2}$$

where d_i is the z-depth of each Gaussian in the viewspace and α_i is the learned opacity multiplied by the projected 2D covariance of the i^{th} Gaussian. In our framework, ground truth depth maps are not required for supervision during training of the 3DGS model $\mathcal{H}$. Using the rendered depth map $\hat{I}_d$, camera intrinsics K, and pose $\hat{p}$, we obtain the 3D coordinate map $X_r^d \in \mathbb{R}^{H\times W\times 3}$ for the rendered image $\hat{I}_r$.

Establishing 2D-3D Correspondences. By combining the 2D-2D correspondences $C_{q,r}$ with the 3D coordinate map X_r^d, we establish 2D-3D correspondences between I_q and the scene. For each matched pixel in I_q, we obtain its corresponding 3D coordinate from X_r^d.

Pose Refinement. Finally, we obtain the refined pose $\hat{p}'$ by feeding these 2D-3D correspondences into a PnP (Gao et al., 2003) solver with RANSAC (Fischler & Bolles, 1981) loop. This process

does not require backpropagation through the pose estimator $\mathcal{F}$ or the 3DGS model $\mathcal{H}$, ensuring efficient computation and enabling its usage with any black-box pose estimator model.

Using 2D-3D correspondences, coupled with PnP + RANSAC, provides a robust pose refinement that is much faster and more accurate than methods relying solely on rendering and comparison (Yen-Chen et al., 2021; Lin et al., 2023; Sun et al., 2023). Furthermore, our method eliminates the requirement of training specialized feature descriptors that previous approaches (Chen et al., 2024a; Moreau et al., 2023; Chen et al., 2022; Zhao et al., 2024) rely on for robustness.

3.3 Faster Alternative with Relative Post Estimation

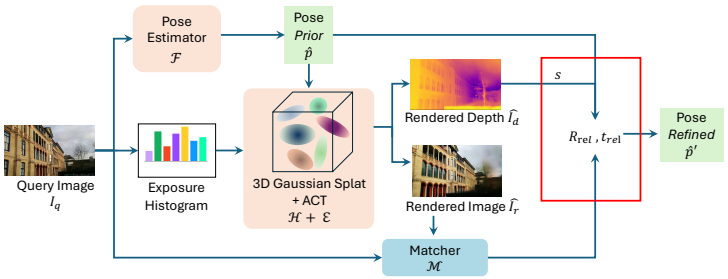


Figure 3: Overview of GS-CPR$_{\text{rel}}$. Different from GS-CPR in Figure 2 (highlight with the red box), we use $\hat{I}_d$ to recover the scale s of $\mathbf{t}_{\text{rel}}$. Then we calculate the refined pose $\hat{p}'$ based on $\mathbf{R}_{\text{rel}}$ and $s\mathbf{t}_{\text{rel}}$ without matching.

While GS-CPR provides high accuracy through 2D-3D correspondences, we also explore an alternative approach that prioritizes computational efficiency. This variant, which we call GS-CPR$_{\text{rel}}$, utilizes MASt3R's point map registration capabilities to estimate relative pose without matching. Figure 3 shows an overview of the GS-CPR$_{\text{rel}}$ approach.

Specifically, MASt3R generates point maps $\mathbf{P}_q$ and $\mathbf{P}_r$ for both the query image I_q and the rendered image $\hat{I}_r$ and predicts the relative rotation $\mathbf{R}_{\text{rel}}$ and translation $\mathbf{t}_{\text{rel}}$ between the two images. However, this relative pose predicted by MASt3R needs to be aligned to the scene's scale s. We recover the scale by aligning the point map $\mathbf{P}_r$ with the depth map $\hat{I}_d$ rendered from the 3DGS model $\mathcal{H}$. The final refined pose $\hat{p}'$ is computed as:

$$\hat{p}' = [\hat{\mathbf{R}}'|\hat{\mathbf{t}}'] = [\mathbf{R}_{\text{rel}}\hat{\mathbf{R}}|\mathbf{R}_{\text{rel}}\hat{\mathbf{t}} + s\mathbf{t}_{\text{rel}}], \tag{3}$$

where $\hat{\mathbf{R}}$, $\hat{\mathbf{t}}$ are the initial rotation and translation estimates. As shown in Table 5 and 6, GS-CPR$_{\text{rel}}$ offers a trade-off between speed and accuracy, making it ideal for rapid refinement of APR methods like DFNet (Chen et al., 2022).

4 Experiments

4.1 Evaluation Setup

Datasets. We evaluate the performance of GS-CPR across three widely used public visual localization datasets. The 7Scenes dataset (Glocker et al., 2013; Shotton et al., 2013) comprises seven indoor scenes with volumes ranging from 1–18 m^3. The 12Scenes dataset (Valentin et al., 2016) features 12 larger indoor scenes, with volumes spanning from 14–79 m^3. The Cambridge Landmarks dataset (Kendall et al., 2015) represents large-scale outdoor scenarios, characterized by challenges such as moving objects and varying lighting conditions between query and training images.

Evaluation Metrics. We report two types of metrics to compare the performance of different methods. The first metric is the median translation and rotation error. The second metric is the recall rate, which measures the percentage of test images localized within a cm and b°.

Baselines. In our experiment, to demonstrate the improvement capabilities of our framework, we use the initial estimates of APR and SCR methods as our baseline. We employ our method on top of

Table 1: Comparisons on 7Scenes dataset. The median translation and rotation errors (cm/°) of different methods. The best results are in **bold** (lower is better). Second best results are indicated with an underline. NRP stands for Neural Rendering-based Pose Estimation.

	Methods	Chess	Fire	Heads	Office	Pumpkin	Redkitchen	Stairs	Avg. ↓ [cm/°]
APR	PoseNet (Kendall et al., 2015)	10/4.02	27/10.0	18/13.0	17/5.97	19/4.67	22/5.91	35/10.5	21/7.74
	MS-Transformer (Shavit et al., 2021)	11/6.38	23/11.5	13/13.0	18/8.14	17/8.42	16/8.92	29/10.3	18/9.51
	DFNet (Chen et al., 2022)	3/1.12	6/2.30	4/2.29	6/1.54	7/1.92	7/1.74	12/2.63	6/1.93
	Marepo (Chen et al., 2024b)	1.9/0.83	2.3/0.92	2.1/1.24	2.9/0.93	2.5/0.88	2.9/0.98	5.9/1.48	2.9/1.04
SCR	DSAC* (Brachmann & Rother, 2021)	**0.5**/0.17	0.8/0.28	0.5/0.34	1.2/0.34	1.2/0.28	**0.7**/0.21	2.7/0.78	1.1/0.34
	ACE (Brachmann et al., 2023)	**0.5**/0.18	0.8/0.33	0.5/0.33	1.0/0.29	1.0/**0.22**	0.8/0.2	2.9/0.81	1.1/0.34
	GLACE (Wang et al., 2024a)	0.6/0.18	0.9/0.34	0.6/0.34	1.1/0.29	**0.9**/0.23	0.8/0.20	3.2/0.93	1.2/0.36
NRP	FQN-MN (Germain et al., 2022)	4.1/1.31	10.5/2.97	9.2/2.45	3.6/2.36	4.6/1.76	16.1/4.42	139.5/34.67	28/7.3
	CrossFire (Moreau et al., 2023)	1/0.4	5/1.9	3/2.3	5/1.6	3/0.8	2/0.8	12/1.9	4.4/1.38
	pNeRFLoc (Zhao et al., 2024)	2/0.8	2/0.88	1/0.83	3/1.05	6/1.51	5/1.54	32/5.73	7.3/1.76
	DFNet + $NeFeS_{50}$ (Chen et al., 2024a)	2/0.57	2/0.74	2/1.28	2/0.56	2/0.55	2/0.57	5/1.28	2.4/0.79
	HR-APR (Liu et al., 2024a)	2/0.55	2/0.75	2/1.45	2/0.64	2/0.62	2/0.67	5/1.30	2.4/0.85
	NeRFMatch (Zhou et al., 2024)	0.9/0.3	1.1/0.4	1.5/1.0	3.0/0.8	2.2/0.6	1.0/0.3	10.1/1.7	2.8/0.7
	MCLoc (Trivigno et al., 2024)	2/0.8	3/1.4	3/1.3	4/1.3	5/1.6	6/1.6	6/2.0	4.1/1.43
	DFNet + **GS-CPR (ours)**	0.7/0.20	0.9/0.32	0.6/0.36	1.2/0.32	1.3/0.31	0.9/0.25	2.2/0.61	1.1/0.34
	Marepo + **GS-CPR (ours)**	0.6/0.18	0.7/0.28	0.5/0.32	1.1/0.29	1.0/0.26	0.8/0.21	1.5/0.44	0.9/0.28
	ACE + **GS-CPR (ours)**	**0.5/0.15**	**0.6/0.25**	**0.4/0.28**	**0.9/0.26**	1.0/0.23	**0.7/0.17**	**1.4/0.42**	**0.8/0.25**

the prevailing APR methods, DFNet (Chen et al., 2022) and Marepo (Chen et al., 2024b), as well as a well-known SCR method, ACE (Brachmann et al., 2023), as the pose estimator $\mathcal{F}$. We follow the default settings of these pose estimators to obtain the initial pose prior for each query image[1]. The term *APR/SCR + GS-CPR* denotes the one-shot refinement. A similar naming convention applies to *APR/SCR + GS-CPR*$_{rel}$. We also include a comparison here with the state-of-the-art NeRF-based methods (Chen et al., 2024a; Moreau et al., 2023; Zhou et al., 2024; Liu et al., 2024a; Germain et al., 2022; Zhao et al., 2024; Liu et al., 2023) and MCLoc (Trivigno et al., 2024), which is a pose refinement framework agnostic to scene representation. MCLoc provides results using 3DGS models as scene representations for the 7Scenes and Cambridge datasets.

Implementation Details. GT Poses: For both the 7Scenes and 12Scenes datasets, we adopt the SfM ground truth (GT) provided by Brachmann et al. (2021). As demonstrated in NeFeS (Chen et al., 2024a), SfM GT can render superior geometric details compared to dSLAM GT for the 7Scenes dataset. Gaussian Splatting: For the training of the 3DGS model of each scene, we utilize the sparse point cloud of training frames generated by COLMAP (Schonberger & Frahm, 2016) as the initial input. We select Scaffold-GS (Lu et al., 2024) as our 3DGS representation, incorporating modifications detailed in Sections 3.1 and 3.2 to adapt exposure and enable depth rendering. Scaffold-GS reduces redundant Gaussians while delivering high-quality rendering compared to the vanilla 3DGS (Kerbl et al., 2023). For the exposure-adaptive ACT module, we follow the default setting in Chen et al. (2024a), computing the query image's histogram in the YUV color space and binning the luminance channel into 10 bins. In addition, we apply temporal object filtering to filter out moving objects in the dynamic scene using an off-the-shelf method (Cheng et al., 2022), leading to better accuracy in scene reconstruction quality and pixel-matching performance. Training Details: We employ the official pre-trained MASt3R (Leroy et al., 2024) model without fine-tuning for 2D-2D matching and resize all images to 512 pixels on their largest dimension. The modified Scaffold-GS model is trained for each scene with 30,000 iterations on an NVIDIA A6000 GPU. We implement our framework with PyTorch (Paszke et al., 2019). Additional details can be found in the Appendix A.1 and A.2.

4.2 Localization Accuracy

We conduct quantitative experiments on three datasets to evaluate the improved localization accuracy of our framework compared to the APR and SCR methods.

7Scenes Dataset. Using the 7Scenes dataset, we evaluate the performance of DFNet, Marepo, and ACE with GS-CPR. Table 1 demonstrates that GS-CPR significantly reduces pose estimation errors for DFNet, Marepo, and ACE with one-shot refinement. Table 2 shows that GS-CPR significantly improves the proportion of query images below 5cm, 5° and 2cm, 2° pose error. It is worth noting

[1]Note that the original paper of Marepo reports results on 7Scenes using dSLAM GT; we retrained the ACE head of Marepo using SfM GT.

Table 2: We report the average percentage (%) of frames below a (5cm, 5°) and (2cm, 2°) pose error across 7Scenes. IR denotes image retrieval.

	Methods	Avg. ↑ [5cm, 5°]	Avg. ↑ [2cm, 2°]
APR	DFNet	43.1	8.4
	Marepo	84.0	33.7
IR+SfM points	HLoc (SP + SG) (Sarlin et al., 2020; 2019)	95.7	84.5
	DVLAD+R2D2 (Torii et al., 2015; Revaud et al., 2019)	95.7	87.2
SCR	DSAC*	97.8	80.7
	ACE	97.1	83.3
	GLACE	95.6	82.2
NRP	DFNet + $NeFeS_{50}$	78.3	45.9
	HR-APR	76.4	40.2
	NeRFMatch	78.4	-
	NeRFLoc (Liu et al., 2023)	89.5	-
	DFNet + **GS-CPR (ours)**	94.2	76.5
	Marepo + **GS-CPR (ours)**	<u>99.4</u>	<u>89.6</u>
	ACE + **GS-CPR (ours)**	**100**	**93.1**

Table 3: Comparisons on Cambridge Landmarks dataset. We report the median translation and rotation errors (cm/°) of different methods. Best results are in **bold** (lower is better) among the NRP approaches.

	Methods	Kings	Hospital	Shop	Church	Avg. ↓ [cm/°]
IR + SfM points	HLoc (SP+SG) (k=1)	13/0.22	18/0.38	6/0.25	9/0.28	12/0.28
	HLoc (SP+SG) (k=10)	11/0.2	15/0.31	4/0.21	7/0.22	9/0.24
APR	PoseNet	93/2.73	224/7.88	147/6.62	237/5.94	175/5.79
	MS-Transformer	85/1.45	175/2.43	88/3.20	166/4.12	129/2.80
	LENS (Moreau et al., 2022)	33/0.5	44/0.9	27/1.6	53/1.6	39/1.15
	DFNet	73/2.37	200/2.98	67/2.21	137/4.02	119/2.90
	PMNet (Lin et al., 2024)	68/1.97	103/1.31	58/2.10	133/3.73	90/2.27
SCR	ACE	29/0.38	31/0.61	5/0.3	19/0.6	21/0.47
	GLACE[1]	20/0.32	20/0.41	5/0.22	9/0.3	14/0.32
NRP	FQN-MN	28/0.4	54/0.8	13/0.6	58/2	38/1
	CrossFire	47/0.7	43/0.7	20/1.2	39/1.4	37/1
	DFNet + $NeFeS_{30}$[2]	37/0.64	98/1.61	17/0.60	42/1.38	49/1.06
	DFNet + $NeFeS_{50}$	37/0.54	52/0.88	15/0.53	37/1.14	35/0.77
	HR-APR	36/0.58	53/0.89	13/0.51	38/1.16	35/0.78
	MCLoc	31/0.42	39/0.73	12/0.45	26/0.8	27/0.6
	DFNet + **GS-CPR (ours)**	23/0.32	42/0.74	10/0.36	27/0.62	26/0.51
	ACE + **GS-CPR (ours)**	**20/0.29**	**21/0.40**	**5/0.24**	**13/0.40**	**15/0.33**

[1] We report the accuracy based on official open-source models (Wang et al., 2024a).
[2] Results of DFNet + $NeFeS_{30}$ taken from Liu et al. (2024a).

that ACE + GS-CPR outperforms HLoc (Superpoint (DeTone et al., 2018) + Superglue (Sarlin et al., 2020)), indicating that 3DGS has the potential to replace traditional point-clouds in visual localization pipelines. Figure 4 (a) shows that after refinement using our GS-CPR, the rendered image of the estimated pose better matches the real image.

Cambridge Landmarks Dataset. We conduct a quantitative evaluation by deploying DFNet and ACE with GS-CPR. Marepo is not included in this comparison due to the absence of an official model for this dataset. Table 3 demonstrates that GS-CPR significantly reduces pose estimation errors for both DFNet and ACE. Specifically, the accuracy of DFNet + GS-CPR with one-shot optimization significantly surpasses that of CrossFire and DFNet + NeFeS with 30 and even 50 steps of optimization (see Table 3). This result fully demonstrates the efficiency of our GS-CPR. On the Kings College scene, DFNet + GS-CPR outperforms ACE after our refinement. ACE + GS-CPR consistently improves ACE accuracy across all four scenes. Refining the pose using our method results in a rendered image that aligns more accurately with the ground truth image as illustrated in Figure 4 (c).

12Scenes Dataset. We conduct the quantitative evaluation using Marepo and ACE with GS-CPR. The former works (Brachmann et al., 2023; Wang et al., 2024a) report the percentage of frames below a 5cm, 5° pose error. Since SCR methods have already achieved good results with this metric, in this paper we use a more stringent standard (2cm, 2°) and report the median translation and

Table 4: We report the average accuracy (%) of frames meeting a $[5\text{cm}, 5°]$, $[2\text{cm}, 2°]$ pose error threshold, and the median translation and rotation errors (cm/°) across 12Scenes.

Methods	Avg. Err ↓ [cm/°]	Avg. ↑ $[5\text{cm}, 5°]$	Avg. ↑ $[2\text{cm}, 2°]$
Marepo	2.1/1.04	95	50.4
DSAC*	**0.5**/0.25	99.8	96.7
ACE	0.7/0.26	**100**	97.2
GLACE	0.7/0.25	**100**	97.5
Marepo + **GS-CPR (ours)**	0.7/0.28	98.9	90.9
ACE + **GS-CPR (ours)**	**0.5/0.21**	**100**	**98.7**

Table 5: We report the average accuracy (%) of frames meeting a $[5\text{cm}, 5°]$ pose error threshold, and the median translation and rotation errors (cm/°).

Datasets	7Scenes		Cambridge
Methods	Avg. Acc ↑ $[5\text{cm}, 5°]$	Avg. Err ↓ [cm/°]	Avg. Err ↓ [cm/°]
DFNet	43.1	6/1.93	119/2.9
DFNet + **GS-CPR$_{\text{rel}}$ (ours)**	80.5	2.7/0.38	55/0.57
DFNet + **GS-CPR (ours)**	**94.2**	**1.1/0.34**	**26/0.51**
ACE	97.1	1.1/0.34	21/0.47
ACE + **GS-CPR$_{\text{rel}}$ (ours)**	79.9	2.8/0.43	47/0.54
ACE + **GS-CPR (ours)**	**100**	**0.8/0.25**	**15/0.33**

rotation errors (cm/°). Table 4 shows that GS-CPR significantly improves the percentage of query images below $2\text{cm}, 2°$ pose error and median pose error for Marepo and ACE. Figure 4 (b) shows that after refinement using our GS-CPR, the rendered image with our pose estimation aligns better with the real image.

GS-CPR vs. GS-CPR$_{\text{rel}}$. We compare GS-CPR, a pose refinement framework that uses 2D-3D correspondence, with GS-CPR$_{\text{rel}}$, a faster alternative that uses relative pose from MASt3R. Both frameworks are evaluated on 7Scenes and Cambridge Landmarks datasets using DFNet and ACE predictions. Table 5 shows that GS-CPR$_{\text{rel}}$ achieves notable accuracy improvement with DFNet on both indoor and outdoor datasets, though it is less effective than GS-CPR. However, GS-CPR$_{\text{rel}}$ is significantly faster than GS-CPR and other NeRF-based methods, as discussed in Section 4.3. While GS-CPR$_{\text{rel}}$ improves coarse pose estimates from APR methods like DFNet, it struggles with accurate pose estimates from SCR methods. For ACE, GS-CPR$_{\text{rel}}$ results in performance degradation because our pose refinement relies on the relative pose estimator MASt3R, which struggles to provide more accurate relative pose estimates when the ACE-predicted pose is sufficiently close to the GT pose. Higher median rotation and translation errors in Table 5 compared to GS-CPR indicate that scale recovery is not the only challenge for GS-CPR$_{\text{rel}}$, as rotation is scale-independent.

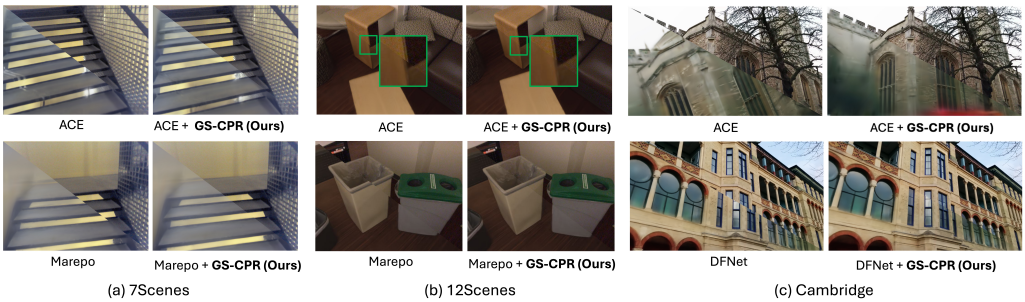


Figure 4: Our GS-CPR enhances pose predictions for Marepo, DFNet, and ACE. Each subfigure is divided by a diagonal line, with the **bottom left** part rendered using the estimated/refined pose and the **top right** part displaying the ground truth image. Patches highlighting visual differences are emphasized with green insets for enhanced visibility.

Table 6: Runtime Analysis (test on Cambridge Landmarks).

Methods	CrossFire	DFNet + NeFeS$_{50}$	HR-APR	MCLoc	DFNet + **GS-CPR$_{\text{rel}}$ (ours)**	DFNet + **GS-CPR (ours)**	ACE + **GS-CPR (ours)**
Avg. ↓ [cm/°]	37/1.0	35/0.8	35/0.8	27/0.6	55/0.6	<u>26</u>/<u>0.5</u>	**15/0.3**
Avg. ↓ time (s)	0.3	10	8.5	2.4	**0.08**	<u>0.18</u>	0.19

Table 7: Results of different matchers (LoFTR (Sun et al., 2021), DUSt3R (Wang et al., 2024b), and MASt3R (Leroy et al., 2024)) on the 7Scenes dataset. GS-CPR $^{\text{L}}$ denotes using LoFTR as the matcher $\mathcal{M}$, GS-CPR $^{\text{D}}$ denotes using DUSt3R as $\mathcal{M}$, and GS-CPR $^{\text{M}}$ denotes using MASt3R as $\mathcal{M}$. The table presents median translation and rotation errors (cm/°) of the different methods.

Methods	Marepo	+ GS-CPR $^{\text{L}}$	+ GS-CPR $^{\text{D}}$	+ GS-CPR $^{\text{M}}$	ACE	+ GS-CPR $^{\text{L}}$	+ GS-CPR $^{\text{D}}$	+ GS-CPR $^{\text{M}}$
Avg. ↓ [cm/°]	2.9/1.04	1.5/0.40	2.1/0.7	**0.9/0.28**	1.1/0.34	1.0/0.31	1.5/0.6	**0.8/0.25**

4.3 Runtime Analysis

We evaluate the runtime of the proposed framework using an NVIDIA GeForce RTX 4090 GPU. On average, 3DGS rendering takes 3.7 ms on the 7Scenes dataset and 12 ms on the Cambridge Landmarks dataset (due to higher scene complexity and image resolution). MASt3R relative pose estimation takes 71 ms. MASt3R matching takes an additional 42 ms, and PnP+RANSAC takes another 52 ms. As a result, our GS-CPR$_{\text{rel}}$ only adds 71 ms to the inference time of the pose estimator $\mathcal{F}$, and our GS-CPR adds less than 180 ms overhead. All time measurements are averaged over 1,000 runs. We compare the runtime and accuracy with other methods in Table 6. On the Cambridge Landmarks dataset, MCLoc requires an average of 2.4 s per query with 80 iterations (Trivigno et al., 2024). In contrast, our ACE+GS-CPR with one-shot optimization only takes 0.19 s per query. Therefore, in terms of efficiency and improvement, our GS-CPR is better than MCLoc when using 3DGS as scene representation. Although GS-CPR$_{\text{rel}}$ is less accurate than GS-CPR, it is more efficient. GS-CPR$_{\text{rel}}$ provides a feasible solution to pose refinement when the time budget is important.

4.4 Ablation Study

In this section, we first demonstrate the rationale behind selecting MASt3R as the matcher $\mathcal{M}$ in GS-CPR. Subsequently, we show that ACT effectively reduces the domain gap between the query image and the rendered image, thereby enhancing the refinement accuracy.

Different Matchers. We compare three matching methods: LoFTR (Sun et al., 2021), DUSt3R (Wang et al., 2024b), and MASt3R – within GS-CPR in the 7Scenes dataset. For DUSt3R and MASt3R, we resize all images to 512 pixels in their largest dimension. For LoFTR, we use the pre-trained model for indoor scenes and maintain the frames in the 7Scenes dataset at 640×480. As shown in Table 7, Marepo + GS-CPR and ACE + GS-CPR using MASt3R as $\mathcal{M}$ achieve the highest improvement. Conversely, ACE + GS-CPR using DUSt3R does not yield any improvement. Marepo + GS-CPR using DUSt3R and Marepo/ACE + GS-CPR using LoFTR show a lower improvement compared to MASt3R. These results validate our choice of design to use MASt3R as a matcher $\mathcal{M}$.

Affine Color Transformation. To enhance the robustness of the 3DGS model in image rendering and to reduce the domain gap between the rendered image and the query image, we incorporated an ACT module into the Scaffold-GS model, as described in Section 3.1. Figure 5 illustrates the improvement in image rendering quality with the ACT module applied. The performance enhancement of GS-CPR from the ACT module is demonstrated in Table 8. On the Cambridge Landmarks dataset, employing the ACT module in the DFNet + GS-CPR setup reduces both the average median translation and rotation errors.

4.5 Discussion

In this section, we provide additional insights and discussions of our design choices.

Replace Feature Descriptors. Given that 3DGS can render high-quality synthetic images $\hat{I}_r$ in real-time, we show that using a pre-trained 3D foundation model, MASt3R, can directly establish accurate 2D-2D correspondences $C_{q,r}$ between I_q and $\hat{I}_r$ with a sim-to-real domain gap. As demon-

Table 8: Ablation study for ACT module on Cambridge Landmarks dataset. We report the median translation and rotation errors (cm/°).

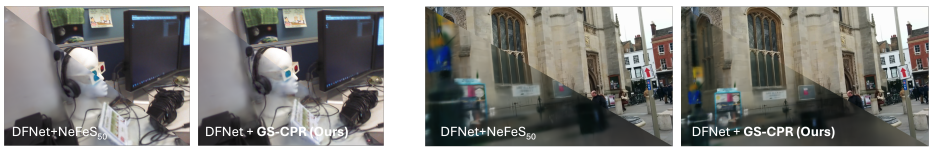
Methods	Kings	Hospital	Shop	Church	Avg. ↓ [cm/°]
DFNet + GS-CPR (w/o. ACT)	34/0.46	54/0.84	12/**0.34**	34/0.72	34/0.59
DFNet + GS-CPR (w. ACT)	**23/0.32**	**42/0.74**	**10**/0.36	**27/0.62**	**26/0.51**

(a) Ground Truth (b) Scaffold-GS/PSNR:16.5 dB (c) Ours/PSNR:18.4 dB

Figure 5: Benefit of the ACT module. A regular 3DGS model tends to render images based on the lighting conditions and the appearance of its training frames, as demonstrated by the synthetic view of Scaffold-GS in (b). However, in challenging visual localization datasets, such as ShopFacade in the Cambridge Landmarks, some query frames may have different exposures compared to the training frames. (c) Our proposed Scaffold-GS + ACT can adaptively adjust the exposure based on the query's histogram.

strated in Section 4.2, GS-CPR achieves significantly higher accuracy than NeRF-based refinement pipelines that rely on feature rendering. Direct RGB matching makes our framework more compact, reduces runtime, eliminates the need for training additional neural radiance features, and simplifies both deployment and usage.

Efficient and Effective Pose Refinement. As a pose estimator, DFNet provides less accurate predictions than Marepo and ACE, but NeFeS reports the best results over DFNet. To ensure a fair comparison with NeFeS, we present examples in Figure 6 illustrating that our GS-CPR outperforms NeFeS in both efficiency and effectiveness. With only one-shot optimization, our GS-CPR achieves higher accuracy than NeFeS with 50 optimization iterations when combined with DFNet on both the indoor 7Scenes and outdoor Cambridge Landmarks datasets. This superior performance is due to our method's leverage of 3D geometry (depth rendering) of the representation, unlike previous NeRF-based refinement methods (Chen et al., 2024a; Yen-Chen et al., 2021) that use only 2D feature/photometric information in an iterative process, rendering candidate poses and comparing them with the target image. Additional discussion can be found in the Appendix A.3.



(a) 7Scenes (b) Cambridge

Figure 6: A comparison between DFNet + GS-CPR and DFNet + NeFeS_{50}.

5 Conclusion

We present GS-CPR, a novel test-time camera pose refinement framework leveraging 3DGS for scene representation to improve the localization accuracy of state-of-the-art APR and SCR methods. GS-CPR enables one-shot pose refinement using only a single RGB query and a coarse initial pose estimate from APR and SCR methods. Our approach outperforms existing NeRF-based optimization methods in both accuracy and runtime across various indoor and outdoor visual localization benchmarks, achieving new state-of-the-art accuracy on two indoor datasets. These results demonstrate the effectiveness and efficiency of our proposed framework.

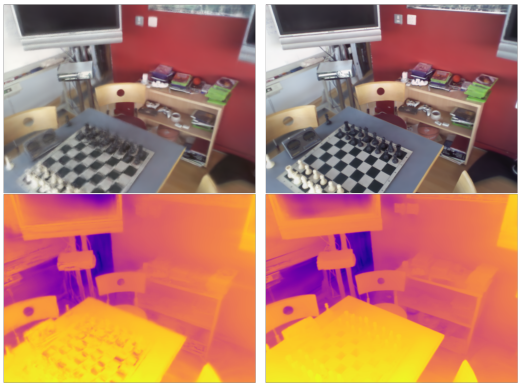
A APPENDIX

A.1 GT POSES DETAILS

In Section 4.2, we report evaluation results based on the SfM ground truth (GT) poses for the 7Scenes dataset, as these poses can render higher quality images (Chen et al., 2024a). Since NeFeS (Chen et al., 2024a) demonstrates the superior accuracy of SfM poses using NeRF as the scene representation, we provide a quantitative comparison in Table 9 and illustrative rendering examples of 3DGS in Figure 7. These results affirm that SfM poses are more accurate, leading to higher quality rendered images and depth maps when using 3DGS. We utilize pre-built COLMAP models from Brachmann et al. (2021) for 7Scenes and 12Scenes datasets, and the models from HLoc toolbox (Sarlin et al., 2019) for the Cambridge landmarks dataset. For the 7Scenes dataset, we enhance the accuracy of the sparse point cloud by utilizing dense depth maps provided by the dataset, combined with the HLoc toolbox and rendered depth maps (Brachmann & Rother, 2021).

Table 9: Quatitative comparison between the 3DGS models implemented in Section 4.1 trained by dSLAM GT poses and SfM GT poses. We report the average PSNR (dB) for the test frames in each scene. The best results are in bold (higher is better).

	dSLAM GT	SfM GT
Scenes	avg. PSNR ↑	avg. PSNR ↑
chess	19.6	**23.1**
fire	19.8	**21.2**
heads	18.4	**19.7**
office	19.4	**21.7**
pumpkin	20.3	**23.2**
redkitchen	18.5	**21.4**
stairs	19.7	**20.1**
avg.	19.4	**21.5**



(a) dSLAM GT (b) SfM GT

Figure 7: Render performance example (dSLAM GT vs. SfM GT). The 3DGS model trained with SfM GT poses (b) renders superior geometric details compared to the dSLAM 3DGS (a) for the same query image, particularly in the chessboard and pieces area.

A.2 SEMANTIC SEGMENTATION WHEN BUILDING 3DGS

To handle challenges in outdoor datasets, we apply temporal object filtering to filter out moving objects in the dynamic scene using an off-the-shelf method (Cheng et al., 2022), leading to better accurate scene reconstruction quality and pixel-matching performance. We show examples of semantic segmentation in Figure 8 and its effect on novel view synthesis (NVS) results in Figure 9. This approach, together with ACT, allows our 3DGS models to provide more robust and better rendering results.

Figure 8: Example of masking on the ShopFacade scene. Top: original images; Bottom: corresponding semantic masks.

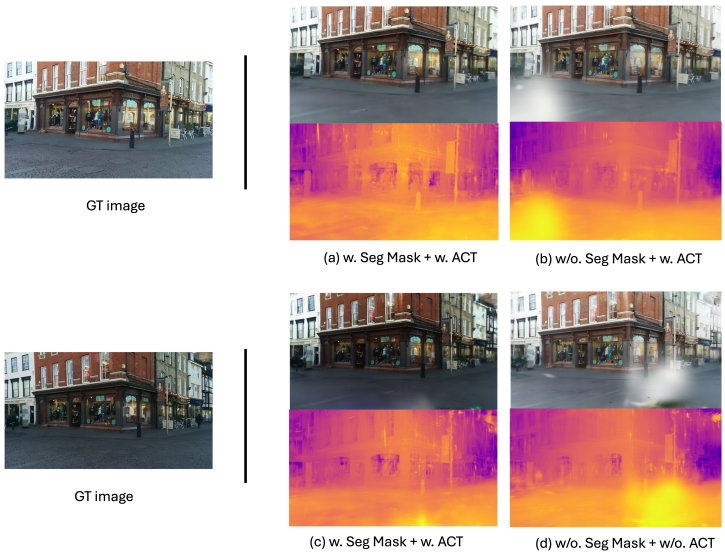


Figure 9: Rendering performance comparison. The 3DGS model trained with segmentation masks renders superior geometric details and fewer artifacts compared to the model trained without masks.

A.3 The advantages of GS-CPR over other approaches

Advantages over render and comapre methods: Methods (Yen-Chen et al., 2021; Lin et al., 2023; Chen et al., 2024a; Sun et al., 2023; Trivigno et al., 2024) leverage only the geometric information of the representation for rendering but do not use it for 2D-3D matching. Consequently, they offer limited accuracy gains and are hindered by slow convergence and high computational costs due to iterative rendering. While NeFeS (Chen et al., 2024a) reduces rendering time and cost by using feature maps and feature loss rather than photometric loss, its accuracy potential remains lower than methods employing 2D-3D matches from original RGB images due to the loss of information in feature maps.

Advantages over structure-based methods: Classical 3D structure-based methods, such as HLoc (Dusmanu et al., 2019; Sarlin et al., 2019; Taira et al., 2018; Noh et al., 2017; Sattler et al., 2016; Sarlin et al., 2020; Lindenberger et al., 2023), estimate camera poses using a 3D SfM point cloud and a reference image database. HLoc requires storing a descriptor database and retrieving the top-k most similar images for 2D-3D correspondences, typically requiring k= 5 to 40 images for robust localization (Humenberger et al., 2022; Sarlin et al., 2022; Leroy et al., 2024). Our approach offers two key advantages: (1) While HLoc requires k matching operations, our GS-CPR only requires one, and its single-shot pose optimization surpasses the accuracy of traditional HLoc. (2) For

challenging queries, even the top-1 retrieved image may have limited overlap with the query (Liu et al., 2024b). However, since GS-CPR performs NVS based on APR and SCR predictions, the rendered images exhibit a greater overlapping region with the query, leading to more accurate matches. We provide examples in Figure 10. The key insight is that both image retrieval and ACE pose-based retrieval are restricted to identifying queries within a limited reference pool. In contrast, our approach theoretically allows for an unlimited reference pool. (3) Using 3DGS instead of sparse point clouds for scene representation enables the domain shift of the rendered image according to the query's exposure through a learning approach, offering greater flexibility.

System design analysis: Our approach goes beyond simply combining 3DGS and MASt3R. As outlined in Section 3.2, our method leverages the matching components of MASt3R to eliminate the need for training extra features to match image pairs with a sim-to-real domain gap—a common limitation of other NeRF-based pose estimation techniques. However, relying solely on MASt3R with reference images fails to deliver accurate metric translation due to the lack of scale information and cannot build 2D-3D matches for absolute pose estimation. This limitation arises because MASt3R is unable to generate metric 3D points within the pre-built global coordinate system. For instance, Jiao et al. (2024) addresses this problem in robotics tasks by incorporating a depth camera. To resolve this challenge, 3DGS in our framework serves a critical function by rendering metric depth, enabling accurate 2D-3D matching. Besides, the rendered view generated by 3DGS from SCR and APR poses aligns much better than normal image retrieval from fixed reference images. This integration is important in recovering precise scale and achieving robust and accurate pose estimation with sufficient matches. By combining the strengths of these components, our framework addresses current limitations.

A.4 Supplementary Visualization

To complement our quantitative analysis, we present additional results in Figure 11 that provide a qualitative perspective on pixel-wise alignment using NVS based on 3DGS across three datasets. A video is also included in the supplementary material.

A.5 Failure cases and Limitation

One limitation of our method lies in its dependency on the accuracy of the initial pose estimates provided by the pose estimator. When the initial pose is highly inaccurate, the overlap between the rendered images and the query image is insufficient to establish reliable 2D-3D correspondences for accurate pose estimation. As shown in Figure 12, GS-CPR cannot refine the DFNet's initial pose in this case because it is too far away from the GT pose.

Following Section 4.5 of NeFeS (Chen et al., 2024a), we conduct quantitative experiments to evaluate the limitations of GS-CPR. Specifically, we introduce random perturbations to the ground truth poses of test frames on the ShopFacade scene, applying fixed magnitudes of rotational and translational errors independently. The results after pose refinement using GS-CPR are presented in Table 10 and Table 11. Our framework can improve the accuracy when rotation error $< 50^\circ$ and translation error < 8 meters, respectively. In contrast, NeFeS achieves accuracy improvements only for rotational errors under 35° and translational errors below 4 meters. These findings highlight that our method significantly expands the optimization range, enhancing its robustness to larger pose perturbations.

Table 10: Average rotation error after refinement by GS-CPR.

Jitter-magnitude (°)	5	10	20	30	40	50	55	60
Avg. Rot. Error (°)	0.23	0.23	0.27	0.35	0.6	7	26	83

Table 11: Average translation error after refinement by GS-CPR.

Jitter-magnitude (m)	1	2	3	4	5	6	8	10
Avg. Trans. Error (m)	0.19	0.38	0.51	0.88	1.13	2.0	3.1	10.7

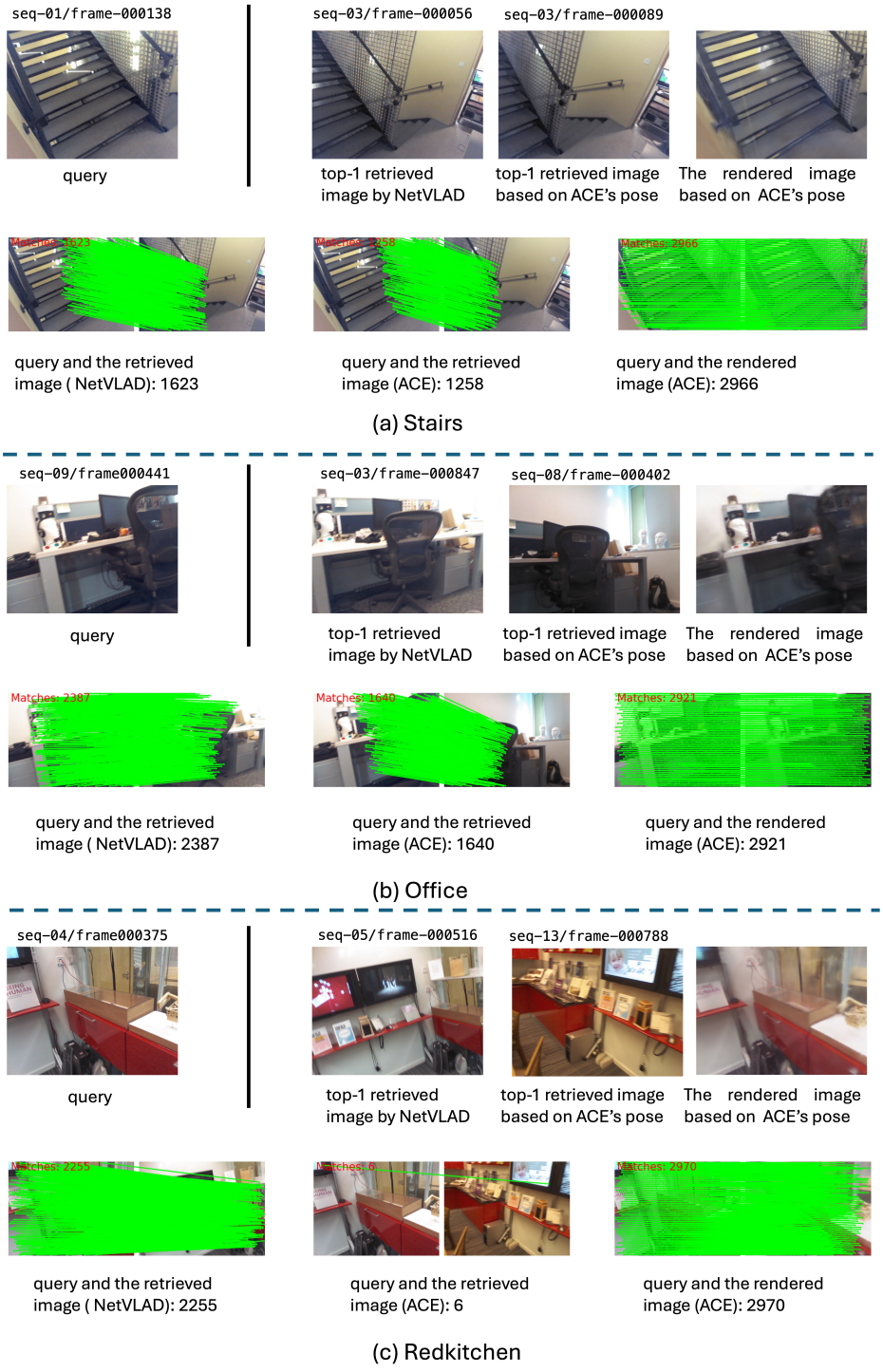


Figure 10: The image rendered from the pose estimator's predictions exhibits a greater overlapping region with the query image than the one retrieved by NetVLAD (Arandjelovic et al., 2016) and the one retrieved by ACE's pose. We use MASt3R as the matcher. Since the matches are very dense, we show the number of matches but only visualize 20% of the matches.

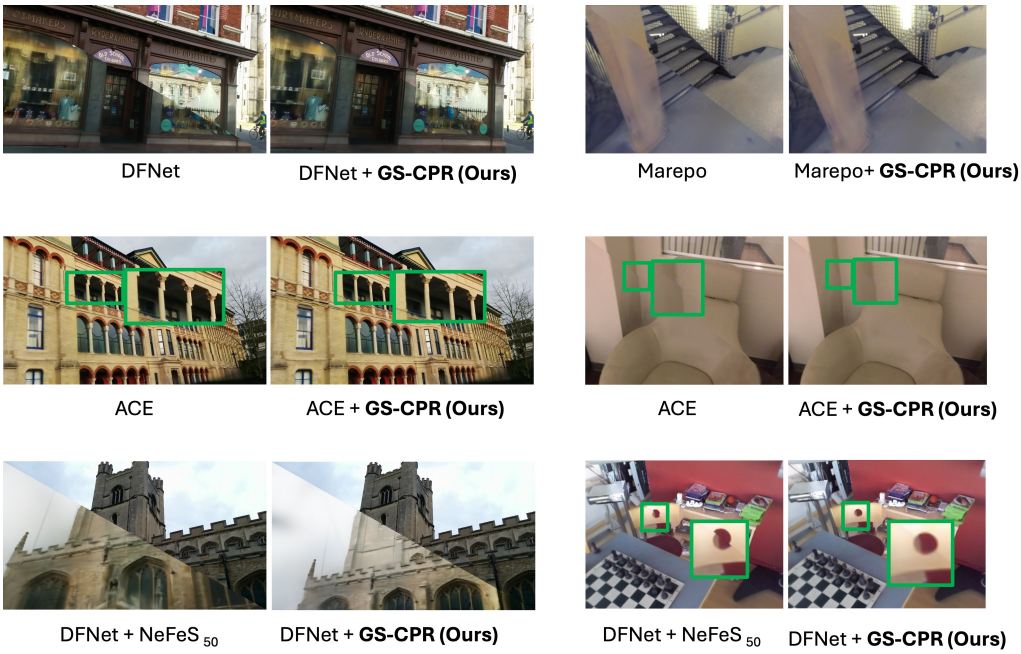


Figure 11: Each subfigure is divided by a diagonal line, with the **bottom left** part rendered using the estimated/refined pose and the **top right** part displaying the ground truth image. Patches highlighting visual differences are emphasized with green insets for enhanced visibility.

Figure 12: Failure case example. Each subfigure is divided by a diagonal line, with the **bottom left** part rendered using the estimated/refined pose and the **top right** part displaying the ground truth image.

This paper demonstrates the effectiveness of our framework on commonly used datasets and benchmarks. However, reconstructing high-quality 3DGS models for large scenes remains a significant challenge. Exploring the application of this framework to large-scale scenes for accurate visual camera relocalization is a promising avenue for future work.

Table 12: We report the average accuracy (%) of frames meeting a $[5\text{cm}, 5^\circ]$, $[2\text{cm}, 2^\circ]$ pose error threshold, and the median translation and rotation errors (cm/°) across 7Scenes and 12Scenes.

Datasets	Methods	Avg. Err ↓ [cm/°]	Avg. ↑ $[5\text{cm}, 5^\circ]$	Avg. ↑ $[2\text{cm}, 2^\circ]$
7Scenes	GLACE	1.2/0.36	95.6	82.2
	GLACE + **GS-CPR (ours)**	**0.8/0.27**	**99.5**	**90.7**
12Scenes	GLACE	0.7/0.25	**100**	97.5
	GLACE + **GS-CPR (ours)**	**0.5/0.21**	**100**	**98.9**

Table 13: Comparisons on Cambridge Landmarks dataset. We report the median translation and rotation errors (cm/°) of different methods.

Methods	Kings	Hospital	Shop	Church	Avg. ↓ [cm/°]
GLACE[1]	20/0.32	20/0.41	5/0.22	9/0.3	14/0.32
GLACE[2]	19/0.3	**17**/0.4	**4/0.2**	9/0.3	**12**/0.3
GLACE + **GS-CPR (ours)**	**17/0.28**	18/**0.34**	5/0.21	**8/0.28**	**12/0.28**

[1] Accuracy based on official open-source models (Wang et al., 2024a).
[2] Accuracy reported in the paper (Wang et al., 2024a).

A.6 Supplementary Experiments

GLACE (Wang et al., 2024a) is an enhanced version of ACE tailored for large-scale outdoor scenes, while exhibiting nearly identical accuracy in indoor environments compared to ACE. We present the results of GLACE + GS-CPR in Tables 12 and 13 to provide supplementary results for evaluating the performance of our approach. GS-CPR significantly improves GLACE accuracies in two of the three datasets (7scenes and 12scenes), demonstrating the effectiveness of our method. On the Cambridge Landmarks dataset, we achieve comparable results, with a slight edge in rotational error.

References

Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5297–5307, 2016.

Matteo Bortolon, Theodore Tsesmelis, Stuart James, Fabio Poiesi, and Alessio Del Bue. 6dgs: 6d pose estimation from a single image and a 3d gaussian splatting model. *arXiv preprint arXiv:2407.15484*, 2024.

Kazii Botashev, Vladislav Pyatov, Gonzalo Ferrer, and Stamatios Lefkimmiatis. Gsloc: Visual localization with 3d gaussian splatting. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5664–5671. IEEE, 2024.

Eric Brachmann and Carsten Rother. Visual camera re-localization from rgb and rgb-d images using dsac. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5847–5865, 2021.

Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6684–6692, 2017.

Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6218–6228, 2021.

Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5044–5053, 2023.

Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-posenet: absolute pose regression with photometric consistency. In *2021 International Conference on 3D Vision (3DV)*, pp. 1175–1185. IEEE, 2021.

Shuai Chen, Xinghui Li, Zirui Wang, and Victor A Prisacariu. Dfnet: Enhance absolute pose regression with direct feature matching. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*, pp. 1–17. Springer, 2022.

Shuai Chen, Yash Bhalgat, Xinghui Li, Jia-Wang Bian, Kejie Li, Zirui Wang, and Victor Adrian Prisacariu. Neural refinement for absolute pose regression with feature synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20987–20996, 2024a.

Shuai Chen, Tommaso Cavallari, Victor Adrian Prisacariu, and Eric Brachmann. Map-relative pose regression for visual re-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20665–20674, 2024b.

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1290–1299, 2022.

Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 224–236, 2018.

Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pp. 8092–8101, 2019.

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*, 24 (6):381–395, 1981.

Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003.

Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-supervising fine-grained region similarities for large-scale image localization. In *European Conference on Computer Vision*, 2020.

Hugo Germain, Daniel DeTone, Geoffrey Pascoe, Tanner Schmidt, David Novotny, Richard Newcombe, Chris Sweeney, Richard Szeliski, and Vasileios Balntas. Feature query networks: Neural surface description for camera pose refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5071–5081, 2022.

Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time rgb-d camera relocalization. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 173–179. IEEE, 2013.

A. Gordo, J. Almazan, J. Revaud, and D. Larlus. End-to-end learning of deep visual representations for image retrieval. *IJCV*, 2017.

Martin Humenberger, Yohann Cabon, Noé Pion, Philippe Weinzaepfel, Donghwan Lee, Nicolas Guérin, Torsten Sattler, and Gabriela Csurka. Investigating the role of image retrieval for visual localization: An exhaustive benchmark. *International Journal of Computer Vision*, 130(7):1811–1836, 2022.

Jianhao Jiao, Jinhao He, Changkun Liu, Sebastian Aegidius, Xiangcheng Hu, Tristan Braud, and Dimitrios Kanoulas. Litevloc: Map-lite visual localization for image goal navigation. *arXiv preprint arXiv:2410.04419*, 2024.

Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *2016 IEEE international conference on Robotics and Automation (ICRA)*, pp. 4762–4769. IEEE, 2016.

Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5974–5983, 2017.

Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 2938–2946, 2015.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pp. 71–91. Springer, 2024.

Jingyu Lin, Jiaqi Gu, Bojian Wu, Lubin Fan, Renjie Chen, Ligang Liu, and Jieping Ye. Learning neural volumetric pose features for camera localization. In *European Conference on Computer Vision*, pp. 198–214. Springer, 2024.

Yunzhi Lin, Thomas Müller, Jonathan Tremblay, Bowen Wen, Stephen Tyree, Alex Evans, Patricio A Vela, and Stan Birchfield. Parallel inversion of neural radiance fields for robust pose estimation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9377–9384. IEEE, 2023.

Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17627–17638, 2023.

Changkun Liu, Shuai Chen, Yukun Zhao, Huajian Huang, Victor Prisacariu, and Tristan Braud. Hr-apr: Apr-agnostic framework with uncertainty estimation and hierarchical refinement for camera relocalisation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8544–8550. IEEE, 2024a.

Changkun Liu, Jianhao Jiao, Huajian Huang, Zhengyang Ma, Dimitrios Kanoulas, and Tristan Braud. Air-hloc: Adaptive retrieved images selection for efficient visual localisation. *arXiv preprint arXiv:2403.18281*, 2024b.

Jianlin Liu, Qiang Nie, Yong Liu, and Chengjie Wang. Nerf-loc: Visual localization with conditional neural radiance field. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9385–9392. IEEE, 2023.

Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20654–20664, 2024.

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 *European Conference on Computer Vision*, pp. 405–421. Springer, 2020.

Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Lens: Localization enhanced by nerf synthesis. In *Conference on Robot Learning*, pp. 1347–1356. PMLR, 2022.

Arthur Moreau, Nathan Piasco, Moussab Bennehar, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Crossfire: Camera relocalization on self-supervised features from an implicit representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 252–262, 2023.

Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, pp. 3456–3465, 2017.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. *Advances in neural information processing systems*, 32, 2019.

Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12716–12725, 2019.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4938–4947, 2020.

Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, and Marc Pollefeys. Lamar: Benchmarking localization and mapping for augmented reality. In *European Conference on Computer Vision*, pp. 686–704. Springer, 2022.

Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1744–1756, 2016.

Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3302–3312, 2019.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.

Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2733–2742, 2021.

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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2930–2937, 2013.

Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8922–8931, 2021.

Yuan Sun, Xuan Wang, Yunfan Zhang, Jie Zhang, Caigui Jiang, Yu Guo, and Fei Wang. icomma: Inverting 3d gaussians splatting for camera pose estimation via comparing and matching. *arXiv preprint arXiv:2312.09031*, 2023.

Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7199–7209, 2018.

Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1808–1817, 2015.

Gabriele Trivigno, Carlo Masone, Barbara Caputo, and Torsten Sattler. The unreasonable effectiveness of pre-trained features for camera pose refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12786–12798, 2024.

Julien Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *2016 Fourth International Conference on 3D Vision (3DV)*, pp. 323–332. IEEE, 2016.

Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. *arXiv preprint arXiv:1909.03557*, 2019.

Fangjinhua Wang, Xudong Jiang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. Glace: Global local accelerated coordinate encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21562–21571, 2024a.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024b.

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 *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1323–1330. IEEE, 2021.

Boming Zhao, Luwei Yang, Mao Mao, Hujun Bao, and Zhaopeng Cui. Pnerfloc: Visual localization with point-based neural radiance fields. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7450–7459, 2024.

Qunjie Zhou, Maxim Maximov, Or Litany, and Laura Leal-Taixé. The nerfect match: Exploring nerf features for visual localization. In *European Conference on Computer Vision*, pp. 108–127. Springer, 2024.

Statement of Authorship for the paper "GS-CPR: Efficient Camera Pose Refinement via 3D Gaussian Splatting" in Chapter 7.

Paper title	GS-CPR: Efficient Camera Pose Refinement via 3D Gaussian Splatting
Authors	Changkun Liu, **Shuai Chen**, Yash Bhalgat, Siyan Hu, Zirui Wang, Ming Cheng, Victor Adrian Prisacariu, Tristan Braud
Publication status	Published
Publication details	International Conference on Learning Representations (ICLR), 2025.

Student Confirmation

Student name	Shuai Chen	
Contribution to the paper	Second-author contribution: • Conception of research ideas • Design of models • Writing and presentation of the paper	
Signature and Date		Apr. 23th 2025

Supervisor Confirmation

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

Supervisor name	Prof. Victor Adrian Prisacariu	
Supervisor comments	The description is accurate	
Signature and Date		Apr. 23th 2025

Part III

Architecture of Absolute Pose Regression

Chapter 8

Map-Relative Pose Regression

This chapter explores the integration of geometric priors into APR architectures. While previous chapters have demonstrated steady improvements in APR accuracy through enhanced training strategies and post-processing techniques, the underlying architectures still lack explicit 3D geometric reasoning. This omission presents a key limitation: state-of-the-art APR models, by disregarding well-established 3D principles, often require extensive scene-specific training, necessitating days-long novel view synthesis. This shortcoming limits the practicality of APRs for rapid deployment, particularly in dynamic or frequently changing environments.

To tackle this challenge, we propose Map-Relative Pose Regression (*marepo*), a pose regression framework that decouples the regressor from individual scenes. Instead, *marepo* uses a scene-agnostic transformer-based network [171, 20, 44, 164] to map the relationship between scene-specific geometric maps and poses.

These geometric priors are obtained through a fast-training SCR model [11], which produces dense 2D-3D correspondences for each scene. The scene-specific geometric representations serve as inputs for the pose regressor, grounding APR predictions in geometry. The newly designed architecture relieves the data hunger issue of conventional APR methods and enables the regressor to generalise across

diverse scenes. While *marepo* retains the end-to-end efficiency of APR, this method enhances the model with structural awareness of the underlying 3D scene. Unlike traditional geometric-based pipelines, it does not rely on traditional iterative solvers or hand-crafted features. Instead, it directly predicts camera poses from the input images.

A key architectural innovation is the Dynamic Positional Encoding mechanism, which embeds both pixel locations and camera intrinsics into the network. This further strengthens the geometric understanding of the model and improves robustness to variations in camera configurations and imaging conditions.

This framework significantly reduces the mapping time for new environments. The pose regressor network is trained once across many scenes and can be deployed to new environments immediately. When optimal performance is needed, the system supports lightweight fine-tuning in just a few minutes. In general, *marepo* provides a versatile, accurate, and scalable APR solution through the integration of geometric priors, and we hope that this work offers a promising future research direction in APR-based visual relocalisation.

The remainder of this chapter presents our original manuscript for our paper, *Map-Relative Pose Regression for Visual Re-Localization*, accepted in *2024 Conference on Computer Vision and Pattern Recognition*.

Map-Relative Pose Regression for Visual Re-Localization

Shuai Chen[1,2] Tommaso Cavallari[1] Victor Adrian Prisacariu[1,2] Eric Brachmann[1]
[1]Niantic [2]University of Oxford

Abstract

*Pose regression networks predict the camera pose of a query image relative to a known environment. Within this family of methods, absolute pose regression (APR) has recently shown promising accuracy in the range of a few centimeters in position error. APR networks encode the scene geometry implicitly in their weights. To achieve high accuracy, they require vast amounts of training data that, realistically, can only be created using novel view synthesis in a days-long process. This process has to be repeated for each new scene again and again. We present a new approach to pose regression, map-relative pose regression (**marepo**), that satisfies the data hunger of the pose regression network in a scene-agnostic fashion. We condition the pose regressor on a scene-specific map representation such that its pose predictions are relative to the scene map. This allows us to train the pose regressor across hundreds of scenes to learn the generic relation between a scene-specific map representation and the camera pose. Our map-relative pose regressor can be applied to new map representations immediately or after mere minutes of fine-tuning for the highest accuracy. Our approach outperforms previous pose regression methods by far on two public datasets, indoor and outdoor. Code is available: https://nianticlabs.github.io/marepo.*

1. Introduction

Today, neural networks have conquered virtually all sectors of computer vision, but there is still at least one task that they struggle with: visual relocalization. What is visual relocalization? Given a set of mapping images and their poses, expressed in a common coordinate system, build a scene representation. Later, given a query image, estimate its pose, i.e. position and orientation, relative to the scene.

Successful approaches to visual relocalization rely on predicting image-to-scene correspondences, either via matching [8, 21, 38–40, 42, 58] or direct regression [4–6, 14, 57], then solving for the pose using traditional and robust algorithms like PnP [18] and RANSAC [17].

Adopting a different perspective, approaches based on pose regression [12, 25, 32, 46] attempt to perform visual relocalization without resorting to traditional pose solving, by using a single feed-forward neural network to infer poses from single images. The mapping data is treated as a training set where the camera extrinsics serve as supervision. Generally, pose regression approaches come in two flavors, but they both struggle with accuracy compared to correspondence-based methods.

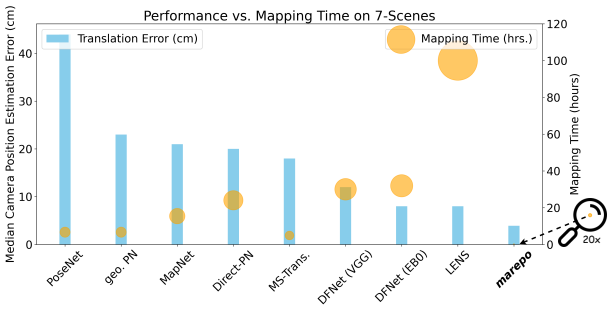


Figure 1. **Camera pose estimation performance vs. mapping time.** The figure shows the median translation error of several pose regression relocalization methods on the 7-Scenes dataset and the time required (proportional to the bubble size) to train each relocalizer on the target scenes. Our proposed approach, ***marepo***, achieves superior performance – by far – on both metrics, thanks to its integration of scene-specific geometric map priors within an accurate, map-relative, pose regression framework.

Absolute pose regression (APR) methods [7, 24, 25] involve training a dedicated pose regressor for each individual scene, enabling the prediction of camera poses to that particular scene. Though the scene coordinate space can be implicitly encoded in the weights of the neural networks, absolute pose regressors exhibit low pose estimation accuracy, primarily due to the often limited training data available for each scene, and struggle to generalize to unseen views [43].

Relative pose regression is a second flavor of pose regression methods [10, 16, 26, 51, 55]. The regressor is trained to predict the relative pose between two images. In a typical inference scenario, the regressor is applied to a pair formed by an unseen query and an image from the mapping set (typically selected via a nearest neighbor-type match-

ing); then, the predicted relative pose can be combined with the known pose of the mapping image to yield the absolute query pose. These methods can be trained on a lot of scene-agnostic data, but their accuracy is still limited: a metric pose between two images can only be predicted approximately [2].

Motivated by those limitations, we propose a new flavor of absolute pose regression: map-relative pose regression (***marepo***). We couple a scene-specific representation – encoding the scale-metric reference space of each target scene – with a general, scene-agnostic, absolute pose regression network. In particular, we utilize a fast-training scene coordinate regression model as our scene representation and train, once and ahead of time, a pose regression network that learns the relationship between a scene coordinate prediction and the corresponding camera pose. This generic relationship allows us to train the pose regressor on hundreds of different scenes, effectively solving the issue of the limited availability of training data afflicting absolute pose regression models. On the other hand, since at localization time our pose regressor is conditioned on a scene-specific map representation, it is able to predict accurate scale-metric poses, unlike relative pose regressors.

Our experiments show that ***marepo*** is a pose regression network with an accuracy on par with structure-based relocalization methods (e.g. [6]), exceeding the accuracy of all other single-frame absolute pose regression methods by far (see Fig. 1). Our scene-agnostic pose regressor can be applied to each new scene representation right away, or (optionally) fine-tuned in just a few minutes for best accuracy. We summarize our main contributions as follows:

1. We propose ***marepo***, a novel Absolute Pose Regression approach that combines a generic and scene-agnostic Map-Relative Pose regression method with a scene-specific metric representation. We show that the network can perform end-to-end inference on previously unseen images and, thanks to the strong and explicit 3D geometric knowledge encoded by the scene-specific component, it can directly estimate accurate, absolute, metric poses.
2. We introduce a transformer-based network architecture that can process a dense set of correspondences between 2D locations in a query image and their corresponding 3D coordinates within the reference system of a previously mapped scene, and estimate the pose of the camera that captured the query image. We further show how a dynamic position encoding applied to the 2D locations in the query image can significantly improve the performance of the method by encoding the intrinsic camera parameters within the transformer input.

2. Related Works

Over the years many efforts in the literature have tackled the problem of visual relocalization, and we have seen a rough demarcation of the types of approach into two main fields: the more traditional approaches, relying on geometric concepts and the estimation of correspondences between images and maps; and the more recent "direct" approaches, relying on neural networks to predict the absolute position and orientation of the image without an intermediate, explicit, matching step linking the 2D image realm with a 3D map of the scene. In the remainder of this section, we briefly explore the main approaches in each of these categories.

2.1. Geometry-based Visual Relocalization

Geometry-based approaches rely on estimating correspondences between pixels in the query images and points in the scene's map. These correspondences effectively establish a 2D-to-3D matching that can be exploited by pose-solving methods such as PnP/RANSAC [17, 18] to compute the pose of the camera at the moment it captures the image.

There are several ways to estimate those correspondences: from classic computer vision approaches using off-the-shelf feature detectors and descriptors to compute matches between image pixels and a database of previously observed 3D points [8, 21, 41, 42]; to more advanced, neural-based approaches that rely on learned descriptors, improved matchers, and different map representations in order to estimate better correspondences from more challenging images or viewpoints [29, 35, 38, 40]. These approaches leverage the underlying geometric principles governing image formation and capture, yielding accurate pose estimations with low errors, often in the order of a few centimeters. However, they are not without a drawback: they generally require the creation of a map (e.g., in the form of a 3D point cloud created via Structure-from-Motion) of the scene ahead of time in order to associate the descriptors to 3D coordinates, and that is typically time-consuming.

In recent years, a new approach to geometry-based relocalization started to become prominent: scene coordinate regression (SCR). In this scenario, the map of the scene is directly encoded in a fixed-size set of weights of a neural network. At localization time, the query image is passed through the network, yielding per-pixel scene coordinates that can be directly used by a pose solver to estimate the camera pose [3–6, 14, 27, 57]. While effective, these methods have typically required training a new network for every new target scene, potentially taking several hours [4], thus hindering their large scale application. Recently, an approach to scene coordinate regression that can take mere minutes to be trained for every scene was presented in [6], making practical deployment of SCR networks a possibility. As the correspondence-based methods mentioned above, coordinate regression approaches are also very accurate by relying on geometric information on the structure of the scene. Nevertheless, scene coordinate regression methods still require an explicit stage where a pose solver has to

process each correspondence generated by the method to estimate the camera pose. Conversely, Absolute Pose Regression methods do not have this requirement since the regressor network can go directly from image to pose in an end-to-end fashion.

2.2. Absolute Pose Regression

Absolute Pose Regression (APR) approaches have also garnered notable attention recently, primarily due to their simplicity and efficiency. These methods directly predict camera poses via end-to-end neural networks. Kendall et al. introduced the first APR approach, named PoseNet [23–25], where a feed-forward neural network directly regresses a 7-dimensional pose vector for every query image. Successive works explore diverse architectural designs such as hourglass networks [30], bifurcated translation and rotation regression [34, 56], attention layers [45–47, 54], and LSTM layers [53]. Other research efforts attempt to improve APR performance with different supervisions, such as a geometric loss [24], relative pose constraints [7], uncertainty awareness [24, 33], or a sequential formulation like temporal filtering [13] and multitasking [37]. Despite these advancements, the accuracy of single-frame-based pose regression remains limited when compared to alternative approaches, such as those based on geometric principles.

Among recent advancements within APR, a promising direction is incorporating novel view synthesis techniques, either by synthesizing large amounts of training data to solve overfitting issues [12, 32, 36] or by integrating them into a fine-tuning process before test time [7, 11, 12]. One drawback of the former is that generating high-quality synthetic data can be a time-intensive process; as for the latter, in addition to time requirements, those approaches typically require extra data from the scene of interest. These limitations pose significant constraints in environments subject to rapid changes, such as those with frequent alterations in furnishings or appearances.

This paper introduces a new category of approach to the pose regression domain, one that tops the need for extensive mapping time, reducing it to mere minutes per scene. It demonstrates enhanced accuracy over previous single-frame APR methods and exhibits rapid scalability to new environments, making it flexible to deploy in a fast-changing world as we live in today.

3. Method

Prevalent pose regression methods are built on top of end-to-end neural network-based approaches. They can be formulated as $\hat{P} = \mathcal{F}(I)$: the camera poses $\hat{P}$ are directly predicted by providing an input image I to the network $\mathcal{F}$. A benefit of this type of approach is its conceptual simplicity and highly efficient inference speed. However, the forward process of typical APR networks – which rely on 2D operations over images and features – does not exploit any 3D geometric reasoning, resulting in insufficient performance compared to state-of-the-art geometry-based methods. In this paper, we propose a first map-relative pose regression approach empowered with explicit 3D geometric reasoning within its formulation, allowing us to regress accurate camera poses while maintaining real-time efficiency and end-to-end simplicity like any other pose regression method.

In the remainder of this section we first give an overview of the transformer-based network architecture we deploy to perform pose regression (Sec. 3.1); then we describe the main components and ideas behind the proposed approach (Sec. 3.2); the loss function optimized during training (Sec. 3.3); and, finally, we show how the scene-agnostic pose-regression transformer can be *optionally* fine-tuned for a specific testing scene in a matter of minutes, thus improving the performance of the method even further compared to a non-fine-tuned regressor (Sec. 3.4).

3.1. Architecture Overview

The main architecture of our method is formed of two components: (1) a CNN-based scene geometry prediction network $\mathcal{G}$ that maps pixels from the input image to 3D scene coordinates; and (2) a transformer-based map-relative pose regressor $\mathcal{M}$ that, given the scene coordinates, estimates the camera poses. Ideally, the network $\mathcal{G}$ is designed to associate each input image to scene-specific 3D information, thus requiring some training process for every new scene processed by the method. Conversely, the map-relative pose regressor $\mathcal{M}$ is a scene-agnostic module trained with large amounts of data and can generalize to unseen maps.

We illustrate our proposed network architecture in Fig. 2. Given an image I from scene S, we pass it to our model which outputs a pose $\hat{P}$. The process is formulated as:

$$\hat{P} = \mathcal{M}(\hat{H}, K) = \mathcal{M}(\mathcal{G}_S(I), K), \tag{1}$$

where $\hat{H} = \mathcal{G}_S(I)$ indicates the image-to-scene coordinates predicted by $\mathcal{G}$ (which was trained ad-hoc for scene S), and $K \in \mathbb{R}^{3\times3}$ is the camera intrinsic matrix associated with the input image. This formulation makes the approach similar to both standard Absolute Pose Regression, in that it generates poses via a feed-forward pass through a neural network, as well as Scene Coordinate Regression since the scene geometry prediction network regresses 3D coordinates directly from each input image. Unlike standard APR, our method has full geometric reasoning on the link between the image and the scene, and, unlike SCR approaches, it does not require a traditional, non-deterministic RANSAC stage to infer the pose. Theoretically, any algorithm capable of predicting 3D scene coordinates from an input image could be a viable candidate as $\mathcal{G}$, since the following transformer we deploy to perform pose regression ($\mathcal{M}$) does not depend on the prior component.

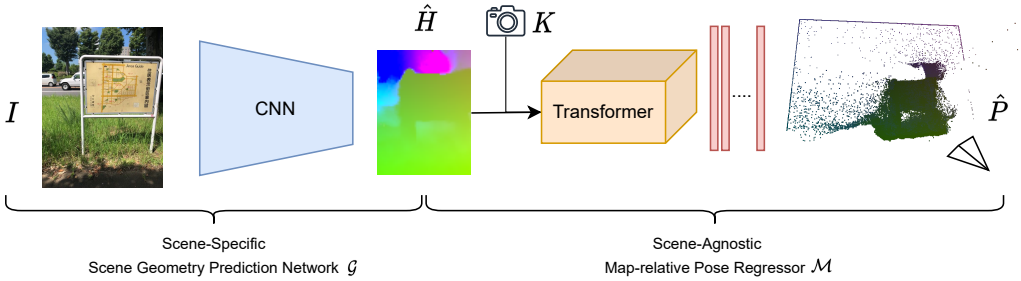


Figure 2. Illustration of the ***marepo*** network. A scene-specific geometry prediction module $\mathcal{G}_S$ processes a query image to predict a scene coordinate map $\hat{H}$. Then, a scene-agnostic map-relative pose regressor $\mathcal{M}$ is used to directly regress the camera pose. Our network's training and inference rely solely on RGB images I and camera intrinsics K without requiring depth information or pre-built point clouds.

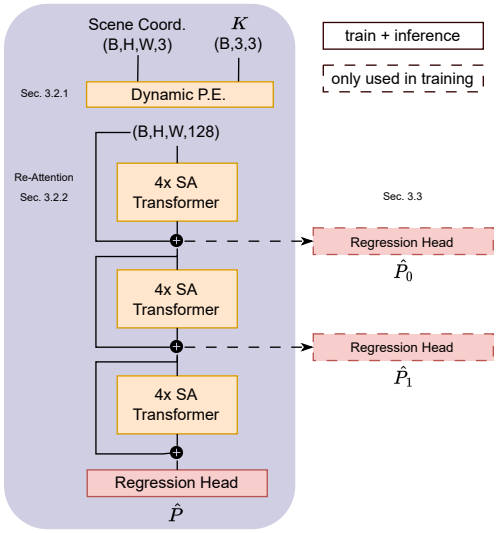


Figure 3. The map-relative pose regressor $\mathcal{M}$ takes as input a tensor of predicted scene coordinate maps and the corresponding camera intrinsics, embeds the information with dynamic positional encoding into higher dimensional features, and finally estimates the camera poses $\hat{P}$. During training, we also predict $\hat{P}_0$ and $\hat{P}_1$ for intermediate supervision.

In the next section, we will focus on detailing the map-relative pose regression network $\mathcal{M}$.

3.2. Map-Relative Pose Regression Architecture

To achieve a robust and scene-agnostic map-relative pose regression, we carefully design the simple yet effective architecture depicted in Fig. 3. The main components of this module are: (a) a novel dynamic positional encoding used to increase the dimensionality of the input scene coordinates – as well as embed their spatial location within the input image – taking into account the intrinsic properties of the camera that captured the frame; (b) several multi-layer self-attention transformer blocks; and finally (c), an MLP-based pose regression head. Given scene coordinate maps $\hat{H}$ (predicted by $\mathcal{G}_S$) and the corresponding camera intrinsic matrices K, the network is able to directly estimate 6-DoF (six degrees of freedom) metric camera poses. We detail the designs of each component in the following sub-sections.

3.2.1 Dynamic Positional Encoding

Unlike many vision transformers (ViTs) for high-level tasks [9, 15, 50], where the transformer is conditioned to operate directly upon input RGB images (or higher-dimensional features), our transformer is designed to interpret accurate 3D geometric information strongly connected to real-world physics. The content a camera captures in its frame is strictly associated with its intrinsic parameters; thus, we propose to use a positional encoding that is conditioned on each individual sensor, allowing us to train the main transformer blocks in a fashion that is generic, i.e., independent of the camera calibration parameters.

Our positional encoding scheme entails the fusion of two different components: (1) a camera-aware 2D positional embedding, associating each predicted scene coordinate to its corresponding pixel location; and (2) a 3D positional embedding that embeds the actual 3D scene coordinate values into a high-frequency domain.

Camera-Aware 2D Positional Embedding We draw inspiration from LoFTR's [50] positional embedding, but integrate information from the camera's intrinsics to generate the high-frequency components that are fed to the network.

Specifically, for each pixel coordinate (u, v) in the input image, we first compute the (x, y) components of the 3D ray originating in the camera center and passing through the pixel (ignoring the z component); then apply the positional embedding from [50] on the (now camera-invariant) ray's directional components. This generates a high-frequency/high-dimensionality embedding, allowing the transformer to correlate the input 3D coordinates (predicted by $\mathcal{G}_S$ and defined in a scene-specific coordinate system) with 3D rays originating from the current camera position, helping with the task of regressing the current camera

pose w.r.t. the origin of the scene coordinate system. Formally, we define the Camera-Aware 2D Positional Embedding as follows:

$$\mathcal{PE}^i_{2D(u,v)} := \begin{cases} \sin\left(\omega_k \cdot X_{\text{ray}}(u)\right), & i = 4k \\ \cos\left(\omega_k \cdot X_{\text{ray}}(u)\right), & i = 4k+1 \\ \sin\left(\omega_k \cdot Y_{\text{ray}}(v)\right), & i = 4k+2 \\ \cos\left(\omega_k \cdot Y_{\text{ray}}(v)\right), & i = 4k+3 \end{cases}, \quad (2)$$

where $\omega_k = \frac{1}{10000^{2k/d}}$ is the frequency band defined for d-dimensional features in which positional encoding is applied on, i is the current feature index, and X_{ray} and Y_{ray} are the X and Y components of the rays passing through (u, v):

$$\begin{aligned} X_{\text{ray}}(u) &= \lambda \frac{u - c_x - \varepsilon}{f_x}, \\ Y_{\text{ray}}(v) &= \lambda \frac{v - c_y - \varepsilon}{f_y}, \end{aligned} \quad (3)$$

with $f_{x/y}$ and $c_{x/y}$ corresponding to the intrinsics of the input frame, $\varepsilon = 0.5$ to achieve zero-mean in the center of the image, and $\lambda = 400$ chosen as a heuristic constant to keep a reasonable numerical magnitude for the final embedding.

3D Positional Embedding We use a 3D positional embedding to map the scene coordinates $p \in \mathbb{R}^3$ predicted by $\mathcal{G}_\mathcal{S}$ to high frequency/dimensionality, inspired by [31]:

$$\begin{aligned} \mathcal{PE}_{3D}(p) = \text{Conv}^d_{3(2m+1)}[p, \sin\left(2^0\pi p\right), \cos\left(2^0\pi p\right), \\ \ldots, \sin\left(2^{m-1}\pi p\right), \cos\left(2^{m-1}\pi p\right)]. \end{aligned} \quad (4)$$

Here, in addition to the sinusoidal embedding mapping the 3D coordinates to a $3(2m+1)$-dimensional space, we also apply a further 1×1 convolution Conv^d_{6m+3} to ensure both $\mathcal{PE}_{2D}$ and $\mathcal{PE}_{3D}$ have the same number of channels.

Fused Positional Embedding Finally, we fuse the 2D and 3D embeddings before passing them to the transformer:

$$\mathcal{PE}_f = \mathcal{PE}_{3D} + \mathcal{PE}_{2D}. \quad (5)$$

3.2.2 Re-Attention for Deep Transformer

As illustrated in the left part of Fig. 3, the core of our map-relative pose regression architecture is formed by twelve self-attention transformers arranged over three blocks of four transformers each. In our implementation, we use Linear Transformers [22] as they reduce the computation complexity of each layer from quadratic to linear in the length of the input (i.e., the resolution of the scene coordinate map).

Since the Dynamic Positional Encoding is fed to the network only at the beginning, we found that the information flow became weaker as the depth of the network increases. To solve this problem, we add what we call a "Re-Attention" mechanism, introducing residual connections every four blocks. Experimentally, we find that this practice is quite effective, allowing the network to converge more quickly and leading to a better generalization.

3.2.3 Pose Regression Head

The last component of the ***marepo*** architecture is a pose regression head. Its structure is simple: first, a residual block formed of three 1×1 convolution layers followed by global average pooling generates a single embedding that represents the whole input scene-coordinate map. Such embedding is then passed to a small MLP (3 layers) that directly outputs the camera pose as a 10-dimensional representation. The pose representation can then be unpacked into translation and rotation: the translation is represented by four homogeneous coordinates (inspired by [6]); the rotation is encoded as a 6D vector representing two un-normalized axes of the coordinate system that are later used to form a full rotation matrix by normalization and cross-product, as in [59].

3.3. Loss Function

The map-relative pose regressor architecture described above is able to directly output a metric pose $\hat{\mathcal{P}}$ (formed of a 3×3 rotation matrix $\hat{R}$, and a translation vector $\hat{\mathbf{t}}$) for each image. In order to train such system we use a standard L1 pose regression loss proposed in [2], defined as follows:

$$\mathcal{L}_{\hat{P}} = \|\hat{R} - R\|_1 + \|\hat{\mathbf{t}} - \mathbf{t}\|_1. \quad (6)$$

Experimentally, we found that adding supervision at intermediate layers of the regressor is beneficial to the overall performance. Therefore, at training time, we additionally apply the pose regression head after each block of four self-attention transformers (see Fig. 3, right) and compute auxiliary losses $\mathcal{L}_{P_0}$ and $\mathcal{L}_{P_1}$ as described above. Thus, the overall loss we optimize during training is as follows:

$$\mathcal{L} = \mathcal{L}_{\hat{P}_0} + \mathcal{L}_{\hat{P}_1} + \mathcal{L}_{\hat{P}}, \quad (7)$$

During inference we only use the last output pose, $\hat{P}$.

3.4. (Optionally) Fine-Tuning the Pose Regressor

As described earlier, the proposed map-relative pose regressor is formed by two main components: an initial scene-specific network $\mathcal{G}_\mathcal{S}$ that is able to predict metric scene coordinates for each pixel (in our implementation, we use an off-the-shelf scene coordinate regression architecture that can be trained in a few minutes for each scene [6]); and a scene-agnostic regressor $\mathcal{M}$ that exploits the geometric information encoded by the scene coordinates to predict the camera pose. The latter is trained once – ahead of time – over a large corpus of data, but it is reusable *as-is* for every new target scene. We find that this hybrid approach works

exceptionally well compared to other APR methods that are trained with the traditional end-to-end image-to-pose protocol, over the course of hours or days.

Still, we also explore whether applying a scene-specific adaptation stage to the transformer-based regressor can be beneficial to the performance of the method. In this scheme, for each new scene being evaluated, after training the scene-specific coordinate regressor $\mathcal{G}_\mathcal{S}$, we fine-tune the pose-regressor $\mathcal{M}$ on the same mapping images, using the same loss as in Sec. 3.3. Fine-tuning the transformer is very efficient in terms of resources required: in the next section we show that, with only two passes over the training dataset for each new scene (taking typically between 1-10 minutes, depending on the number of frames), our method can further improve its performance from what was already state-of-the-art compared to pose regression-based methods.

Notably, the final fine-tuning step is completely *optional* in our approach: the pre-trained $\mathcal{M}$ is already capable of predicting accurate camera poses, given 3D scene coordinates predicted by the geometric network module $\mathcal{G}_\mathcal{S}$.

4. Experiments

4.1. Implementation Details

In this section, we provide details of the process used to train the ***marepo*** network. We first detail the generation of training data, then describe the architectural configuration.

Training Data Generation In our implementation, we employ the Accelerated Coordinate Encoding architecture and training protocol proposed in [6] as $\mathcal{G}$ to train the scene-specific geometry prediction network $\mathcal{G}_\mathcal{S}$ for each scene S in the training dataset, since that allows the training of scene-specific coordinate-regression networks in a speedy fashion ($\sim$5 minutes for each new scene). To train $\mathcal{M}$, we use 450 scenes (indexing from 0 to 459, excluding 200-209) from the Map-Free Dataset [2]. Each image in the dataset has an associated ground truth camera pose computed by the authors of the dataset via SfM [44]. The data includes around 500K frames, with each scene containing images scanning a small outdoor location. Frames from each scan have been split into mapping and query, with $\approx$ 500 mapping frames and $\approx$ 500 queries. In practice, we train 900 scene-specific coordinate regressors $\mathcal{G}_\mathcal{S}$ using frames from each scene and its two splits. Given the efficient scaling capability of the method, we are able to generate a vast amount of 2D-3D correspondences between image pixels and scene coordinates by applying each $\mathcal{G}_\mathcal{S}$ to the frames of the unseen split of its corresponding scan. We use data augmentation during the generation of the correspondences, processing 16 variants of each frame. Specifically, we apply random image rotations of up to 15°; rescale each frame to $0.67 \sim 1.5$ times its original resolution; and, finally, extract random crops. We save the scene coordinate maps output of the preprocessing, together with their corresponding masks indicating which pixels are valid after rotation, the augmented intrinsics, and camera poses. This forms the fixed dataset we use to train the map-relative pose regressor $\mathcal{M}$.

Additionally, we perform online data augmentation by randomly jittering the scene coordinates by $\pm 1m$ and rotating them by up to 180° to further increase data diversity. Note that the random jittering is applied image-wise, i.e., all scene coordinates of an input frame are perturbed by the same transform. We do this to avoid overfitting, ensuring that the network $\mathcal{M}$ does not learn an absolute pose for each frame but rather a pose relative to the scene coordinates.

Network Configuration The scene-specific networks $\mathcal{G}_\mathcal{S}$ process images having the shortest side 480 pixels long and output dense scene coordinate maps with 8x smaller resolution. The map-relative pose regressor $\mathcal{M}$ is built upon a cascade of linear-attention transformer blocks [50] with $d_{model} = 256$ and $h = 8$ parallel attention layers. For the 3D position embedding, we prudently choose $m = 5$ frequency bands due to presence of potentially noisy input.

Training and Hardware Details We train $\mathcal{M}$ using 8 NVIDIA V100 GPUs with a batch size of 64. We use the AdamW [28] optimizer with a learning rate between $3e^{-4}$ to $2e^{-3}$ with a 1-cycle scheduler [49]. The model is trained for $\approx$ 10 days, iterating through the dataset for 150 epochs.

During inference our entire model, including the scene-specific network $\mathcal{G}_\mathcal{S}$ and the map-relative pose regressor $\mathcal{M}$, requires only one GPU, and can estimate camera poses with a real-time throughput, as later shown in Tab. 2.

4.2. Quantitative Evaluation

In the following paragraphs we show the performance of ***marepo*** on two public datasets: one depicting indoor scenes, and one outdoor. We show that the proposed map-relative pose regressor module $\mathcal{M}$ can generalize its predictions to previously unseen scenes, thanks to the scene-specific geometry prediction network $\mathcal{G}_\mathcal{S}$ providing it with 2D-3D correspondences in each scene's metric space.

7-Scenes Dataset We first evaluate our method on the Microsoft 7-Scenes dataset [19, 48], an indoor relocalization dataset that provides up to 7000 mapping images per scene. Each scene covers a limited area (between $1m^3$ and $18m^3$); despite that, previous APR methods require tens of hours or even several days [32] to train a model to relocalize in them. This is nonideal in a practical scenario as the appearance of the scene might have changed within that time frame, thus rendering the trained APR out of date. Conversely, ***marepo*** requires only minutes of training time ($\approx$ 5) for each new scene to generate a geometry-prediction network

Table 1. **Re-localization results on the indoor 7-Scenes dataset.** Pose errors are shown as median translation (cm) and rotation (°) errors. Numbers in **bold** represent the best performance among the APR-based approaches. ***marepo*** denotes our model with generic transformer-based pose regressor $\mathcal{M}$. ***marepo***$_S$ reports the performance of the model after $\mathcal{M}$ has been fine-tuned for each scene.

	Methods	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Average	Mapping Time
SCR	DSAC* [4]	1.9/1.11	1.9/1.24	1.1/1.82	2.6/1.18	4.2/1.41	3.0/1.70	4.2/1.42	2.7/1.41	Hours
	ACE [6]	1.9/0.7	1.9/0.9	0.9/0.6	2.7/0.8	4.2/1.1	4.2/1.3	3.9/1.1	2.8/ 0.93	**5 Minutes**
APR	PoseNet(PN)[25]	32/8.12	47/14.4	29/12.0	48/7.68	47/8.42	59/8.64	47/13.8	44/10.4	Hours
	PN Learn σ^2[24]	14/4.50	27/11.8	18/12.1	20/5.77	25/4.82	24/5.52	37/10.6	24/7.87	Hours
	geo. PN[24]	13/4.48	27/11.3	17/13.0	19/5.55	26/4.75	23/5.35	35/12.4	23/8.12	Hours
	LSTM PN[53]	24/5.77	34/11.9	21/13.7	30/8.08	33/7.00	37/8.83	40/13.7	31/9.85	Hours
	Hourglass PN[30]	15/6.17	27/10.8	19/11.6	21/8.48	25/7.01	27/10.2	29/12.5	23/9.53	Hours
	BranchNet[56]	18/5.17	34/8.99	20/14.2	30/7.05	27/5.10	33/7.40	38/10.3	29/8.30	Hours
	MapNet[7]	8/3.25	27/11.7	18/13.3	17/5.15	22/4.02	23/4.93	30/12.1	21/7.77	Hours
	Direct-PN[11]	10/3.52	27/8.66	17/13.1	16/5.96	19/3.85	22/5.13	32/10.6	20/7.26	Days
	MS-Transformer[47]	11/4.66	24/9.60	14/12.2	17/5.66	18/4.44	17/5.94	17/5.94	18/7.28	Hours
	DFNet (VGG) [12]	5/1.88	17/6.45	6/3.63	8/2.48	10/2.78	22/5.45	16/3.29	12/3.71	Days
	DFNet (EB0) [12]	3/1.15	9/3.71	8/6.08	7/2.14	10/2.76	9/2.87	11/5.58	8/3.47	Days
	LENS [32]	3/1.3	10/3.7	7/5.8	7/1.9	8/2.2	9/2.2	14/3.6	8/3.00	Days
	marepo (Ours)	2.6/1.35	2.5/1.42	2.3/2.21	3.6/1.44	4.2/1.55	5.1/1.99	6.7/1.83	3.9/1.68	**5 Minutes**
	marepo$_S$ (Ours)	**2.1/1.24**	**2.3/1.39**	**1.8/2.03**	**2.8/1.26**	**3.5/1.48**	**4.2/1.71**	**5.6/1.67**	**3.2/1.54**	$\leq$ 15 Minutes

Table 2. **Pose accuracy comparison on the outdoor Wayspots dataset.** Results are reported as the percentage of frames below $10cm/5°$ and $0.5m/5°$ pose error. The map-relative pose regressors $\mathcal{M}$ of our ***marepo***$_S$ experiment are fine-tuned in $\approx$ 1 minute for each scene.

Scene	DSAC* [4]	ACE [6]	PN [23–25]	MST [47]	***marepo*** Ours	***marepo***$_S$ Ours
Throughput (fps)	17.9	17.9	**166.7**	28.4	55.6	55.6
Bears	82.6%/91.6%	80.7%/92.6%	12.9%/35.7%	0.5%/12.8%	**80.7%**/99.3%	**80.7%/99.5%**
Cubes	83.8%/98.1%	97.0%/98.1%	0.0%/0.4%	0.00%/9.9%	**72.4%/96.9%**	71.8%/**96.9%**
Inscription	54.1%/69.7%	49.0%/69.6%	1.1%/6.3%	1.3%/9.7%	**37.8%/74.2%**	37.1%/74.1%
Lawn	34.7%/38.0%	35.8%/38.5%	0.0%/0.2%	0.0%/0.0%	32.6%/**41.6%**	**34.2%**/41.1%
Map	56.7%/87.1%	56.5%/84.7%	14.9%/49.1%	5.6%/25.7%	53.9%/87.7%	**55.1%/87.9%**
Square Bench	69.5%/97.9%	66.7%/97.8%	0.0%/3.0%	0.0%/0.0%	68.6%/**100%**	**70.7%/100%**
Statue	0.0%/0.0%	0.0%/0.0%	0.0%/0.0%	0.0%/0.0%	0.0%/0.0%	0.0%/0.0%
Tendrils	25.1%/26.5%	34.9%/36.8%	0.0%/0.0%	0.9%/23.6%	27.9%/33.4%	**29.3%/34.8%**
The Rock	100%/100%	100%/100%	24.2%/77.5%	10.7%/52.6%	98.1%/**100%**	**99.8%/100%**
Winter Sign	0.2%/5.7%	1.0%/7.6%	0.0%/0.0%	0.0%/0.0%	0.0%/**0.7%**	0.0%/0.3%
Average	50.7%/61.5%	52.2%/62.6%	5.3%/17.2%	1.9%/13.4%	47.2%/63.4%	**47.9%/63.5%**

$\mathcal{G}_\mathcal{S}$ specifically tuned for the target environment. We compare our method with prior Pose Regression approaches in Tab. 1, showing that ***marepo*** is not only a partly scene-agnostic approach that enjoys the fastest mapping time of all APR-based methods, but also obtains $\approx 50\%$ better average performance (in terms of median error). We also show the performance of a fine-tuned variant of our method, ***marepo***$_S$, where, in addition to training the scene-specific $\mathcal{G}_\mathcal{S}$ scene coordinate predictor, we also use the mapping frames to run two epochs of fine-tuning on the $\mathcal{M}$ regressor (see Sec. 3.4). The fine-tuned model ***marepo***$_S$ achieves further improvements in average performance, requiring only between $1.5 \sim 10$ extra minutes of training time, resulting in the only single-frame pose regression-based method able to achieve a similar level of accuracy as one of the current best 3D geometry-based methods, while being more efficient in terms of computational resources required.

Wayspots Dataset We further evaluate our method on the Wayspots dataset [2, 6], which depicts challenging outdoor scenes that even current geometry-based methods struggle with. The dataset contains scans of 10 different areas with associated ground truth poses provided by a visual-inertial odometry system [1, 20]. In Tab. 2 we show a comparison of the performance of the proposed ***marepo*** (as well as the ***marepo***$_S$ models, fine-tuned on the mapping frames of each scene) with two APR-based approaches we reproduced; we also include a comparison with two scene coordinate regression approaches: DSAC* [4] and the current state-of-the-art on Wayspots, ACE [6]. ***marepo*** significantly outperforms previous APR-based methods – such as PoseNet [25] and MS-Transformers [47] – that require on average several hours of training time, and compares favorably with geometry-based methods. We show, for the first time, that an end-to-end image-to-pose regression method relying on

Model	Accuracy	
	$5cm/5°$	$10cm/5°$
Full Architecture (***marepo***)	**16.6%**	**39.6%**
- Re-Attention	10.9%	28.3%
- Dynamic P.E.	3.9%	18.6%

Table 3. We gradually remove *Re-Attention* and *Dynamic Positional Encoding* and report the % of frames relocalized within $5cm/5°$ and $10cm/5°$.

# T Blocks	d_{model}	Accuracy	
		$5cm/5°$	$10cm/5°$
4	128	5.8%	22.2%
8	128	14.7%	39.1%
12	128	16.6%	39.6%
12	256	**19.0%**	**43.5%**

Table 4. Effect on performance of different dimensionality choices in the pose regressor's model. *# T Blocks* denotes the number of transformer blocks used in the model. d_{model} denotes the width of the transformer layers.

geometric priors can achieve a similar level of performance as methods that require the deployment of a (slower) robust solver to estimate the camera pose from a set of potentially noisy 2D-3D correspondences. More specifically, ***marepo*** requires only five minutes to train a network encoding the location of interest within the weights of the $\mathcal{G}_\mathcal{S}$ scene-specific coordinate regressor and (*optionally*) approximately one minute to fine-tune the map-relative regressor $\mathcal{M}$ (as the Wayspot scans have significantly less frames then the 7-Scenes scenes above). At inference time ***marepo*** (or its fine-tuned variant) can perform inference at ≈ 56 frames per second, making it not only accurate, but also extremely efficient in comparison to other methods.

4.3. Ablation Experiments

In the following, we provide additional insights into the design choices adopted whilst designing our method.

Architecture Ablation We run several controlled experiments to justify our architectural design. Note that, for the experiments in this subsection, we trained the transformer-based pose regressor for 50 epochs instead of 150 as in the main experiments. This allows us to complete each experiment in approximately two days without affecting the relative ranking of results in the ablations. For the first experiment, see Tab. 3, we train a smaller transformer $\mathcal{M}$ (with $d_{model} = 128$) and gradually remove the *Re-Attention* and *Dynamic Positional Encoding* components to evaluate their impact on the performance of the Wayspots dataset. The pose accuracy is shown as the percentage of frames relocalized within $5cm/5°$ and $10cm/5°$ error. The table shows consistent degradation without the proposed components.

Next, we show the impact of diverse model configurations by deploying different numbers of transformer blocks and d_{model} dimensions in Tab. 4. We experimented with training even larger models with $d_{model} = 512$ and 16/20 transformer blocks, but found they necessitated substantially more GPU resources and time. We therefore cannot recommend them given the performance-time trade-offs.

Model	Accuracy	
	$10cm/5°$	$50cm/5°$
Per-scene ***marepo***	0.7%	6.0%
Per-scene $\mathcal{M}$	2.9%	18.7%
marepo (Ours)	**47.2%**	**63.4%**

Table 5. Effect of different training strategies. Per-scene ***marepo*** trains the entire network from scratch for every new mapping scene. In per-scene $\mathcal{M}$, we train only the regressor from scratch, on top of a pre-trained $\mathcal{G}_\mathcal{S}$. Finally, ***marepo*** is trained over the entire training dataset and can generalize well to unseen scenes.

Per-Scene Training The proposed pose regressor component $\mathcal{M}$ is designed to be scene-agnostic and has been trained on a large corpus of data. Still, we are interested in evaluating its performance when trained ad-hoc for individual scenes, similar to existing APR-based methods. We conduct two experiments where, instead of using the full training set, we train scene-specific models using mapping sequences from the Wayspots dataset, as shown in Tab. 5. First, the models are trained from scratch, i.e., both the scene geometry regressor $\mathcal{G}_\mathcal{S}$ and the map-relative pose regressor $\mathcal{M}$ are trained as a single entity, similar to PoseNet [25]. The results show extremely poor performance, likely due to the model's inability to learn explicit 3D geometry relations of the scene. For the second variant, we assume a pre-trained $\mathcal{G}_\mathcal{S}$ is provided, then train $\mathcal{M}$ from scratch for each scene. We see results in a similar order of magnitude as other APR methods, such as PoseNet [25] or MST [47] (cf. Tab. 2); still, this training approach performed quite poorly compared to the full ***marepo*** model, where $\mathcal{M}$ is trained on a large-scale dataset to predict truly generic and scene-independent map-relative poses.

5. Conclusion

In conclusion, our paper introduces ***marepo***, a novel approach in Pose Regression that combines the strengths of a scene-agnostic pose regression network with a strong geometric prior provided by a fast-training scene-specific metric representation. The method addresses the limitations of previous APR techniques, offering both scalability and precision in predicting accurate scale-metric poses across diverse scenes. We demonstrate ***marepo***'s superior accuracy and its capability for rapid adaptation to new scenes compared to existing APR methods on two datasets. Additionally, we show how integrating the transformer-based network architecture with dynamic positional encoding ensures robustness to varying camera parameters, establishing ***marepo*** as a versatile and efficient solution for regression-based visual relocalization.

Map-Relative Pose Regression for Visual Re-Localization

Supplementary Material

A. Supplementary Ablations

A.1. Impact of auxiliary losses

In Tab. 6 we present an analysis of the impact of incorporating auxiliary losses $\mathcal{L}_{\hat{P}_0}, \mathcal{L}_{\hat{P}_1}$ into our model training protocol, contrasted with the model devoid of such losses. As mentioned in Section 3.3 of the main paper, we found this implementation beneficial to the overall pose regression performance.

Architecture	Accuracy $10cm/5°$	Accuracy $0.5m/5°$
marepo w/ auxiliary losses (Ours)	**47.2%**	**63.4%**
marepo w/o auxiliary losses	45.7%	62.2%

Table 6. Performance of ***marepo*** trained with and without auxiliary losses as in Equation 7 of the main paper.

A.2. Impact of rotation representation: 9D SVD orthogonalization vs. 6D. Gram-Schmidt

Additionally, we investigated the effects of utilizing alternative rotation representations on our model's performance. For example, Levinson *et al.* [26] demonstrated that SVD orthogonalization facilitates a continuous mapping of a 9D representation onto SO(3), potentially improving pose prediction accuracy beyond that achievable with a 6D representation [59] used in our model. We replaced our 6D with the 9D representation and trained the full ***marepo*** models to assess the differences. The findings indicate that, within our model's framework, the prediction accuracy for 9D rotations marginally lags behind that of 6D rotations (Tab. 7), thereby verify our design choice in the paper.

Accuracy	***marepo*** (9D)	***marepo*** (6D)
$10cm/5°$	46.8%	**47.2%**

Table 7. Ablation on rotation representations using the ***marepo*** model. Accuracies are reported on Wayspots.

A.3. Impact of the SCR component

We use two methods to study the impact of the choice of Scene Coordinate Regression component on the pose estimation performance. First, we replaced the pretrained ACE backbone with a VGG network, then retrained the scene-specific SCRs. The SCRs' outputs were then passed to our pretrained pose regressor $\mathcal{M}$. As indicated in Tab. 8, the choice of SCR does indeed affect the pose regressor's accuracy. However, ***marepo*** also displays robustness to the quality of the input scene coordinates, as the overall performance degradation is not large, demonstrating the capability of our approach to predict accurate poses from scene coordinates generated by different means.

Furthermore, we performed quantitative experiments adding random noise to the scene coordinates passed to $\mathcal{M}$. Specifically, we applied randomly generated noise of different magnitudes (up to 10cm, and up to 50cm) to a variable proportion of the scene coordinates. We show that ***marepo*** is able to cope with large proportions of errors in the input coordinates, without significant drops in performance (up to 60% of the coordinates can be perturbed with 10cm noise, and up to 40% for 50cm noise) (see Tab. 9).

Accuracy	ACE backbone SCR + $\mathcal{M}$	VGG backbone SCR + $\mathcal{M}$
$10cm/5°$	47.2%	46.0%

Table 8. Effect of different scene coordinate regression backbones on the accuracy of the downstream regressor $\mathcal{M}$ on Wayspots.

SCR Noise	0%	20%	40%	60%	80%	100%
$10cm$	47.2%	46.9	46.0	44.5	38.2	26.9
$50cm$	47.2%	46.3	43.2	21.1	10.3	0.3

Table 9. Effect of increasing amounts of random noise applied to the SCR predictions. The top row indicates the proportion of the pixels in each scene coordinate map affected by uniform noise with maximum value indicated at the beginning of each row. We report the $10cm/5°$ accuracy on Wayspots.

B. Supplementary Video

To complement our quantitative analysis, we provide a supplementary video offering a qualitative perspective, primarily focusing on visually comparing the predicted camera trajectories. The trajectories are superimposed on point clouds rendered from the respective scenes, providing an intuitive understanding of each method's performance.

The first segment of the video showcases a comparative analysis of our approach against other open-sourced APR-based methods on the 7-Scenes dataset [19, 48], where we compare vs. PoseNet[24, 25] and DFNet(EB0)[12]; and the Wayspots dataset [2, 6], where we compare vs. PoseNet and MS-Transformer[47], note that both train their models in under one day. For PoseNet, we have utilized the PyTorch implementation provided by Chen *et al.*[11].

Moreover, in the second segment of the video, we show that ***marepo*** compares well qualitatively with the Accelerated Coordinate Encoding[6] structure-based method. This comparison demonstrates that our method achieves similar accuracy to ACE, with the added benefit of producing smoother trajectory estimations in certain scenarios. Notably, our approach provides a faster throughput during inference, underscoring its practical applicability in demanding scenarios.

C. Experiments on 12-Scenes dataset

We show experimental results on the 12-Scenes dataset [52] in Tab. 10. We compare ***marepo*** to the baseline APR methods PoseNet and MS-Transformer. Since the original PoseNet code was implemented on Caffe, we used the open-sourced code from [11]. The results show that ***marepo*** significantly outperforms the baseline APRs, which is consistent with the behavior shown in the main paper compared to the benchmark APR approaches.

Scene	PoseNet	MST	***marepo***
Apt.1 Kitchen	14.3%	3.4%	98.0%
Apt.1 Living	11.2%	9.7%	98.6%
Apt.2 Bed	18.1%	2.9%	96.0%
Apt.2 Kitchen	38.6%	13.8%	100%
Apt.2 Living	13.5%	4.6%	99.7%
Apt.2 Luke	9.1%	4.8%	89.4%
Office 1 Gates 362	34.5%	14.0%	97.2%
Office 1 Gates 381	8.1%	4.1%	84.6%
Office 1 Lounge	17.1%	14.1%	93.9%
Office 1 Manolis	13.7%	8.9%	94.8%
Office 2 Floor 5a	5.2%	1.4%	90.5%
Office 2 Floor 5b	5.2%	7.2%	83.5%
Average	13.5%	7.4%	**93.9%**
median error	9.4cm/3.9°	11.1cm/5.5°	**2.6cm/1.3°**

Table 10. Performance on the 12-Scenes [52] dataset. The accuracy is reported as percentage of query frames localized within $5cm/5°$.

References

[1] Apple. ARKit. Accessed: 26 March 2024. 7

[2] 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, 5, 6, 7, 9

[3] Eric Brachmann and Carsten Rother. Neural- Guided RANSAC: Learning where to sample model hypotheses. In *ICCV*, 2019. 2

[4] Eric Brachmann and Carsten Rother. Visual camera relocalization from RGB and RGB-D images using DSAC. *IEEE TPAMI*, 2021. 1, 2, 7

[5] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC - Differentiable RANSAC for Camera Localization. In *CVPR*, 2017.

[6] 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, 5, 6, 7, 9, 10

[7] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-Aware Learning of Maps for Camera Localization. In *CVPR*, 2018. 1, 3, 7

[8] Federico Camposeco, Andrea Cohen, Marc Pollefeys, and Torsten Sattler. Hybrid scene compression for visual localization. In *CVPR*, 2019. 1, 2

[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 4

[10] Kefan Chen, Noah Snavely, and Ameesh Makadia. Wide-baseline relative camera pose estimation with directional learning. In *CVPR*, 2021. 1

[11] Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-PoseNet: Absolute pose regression with photometric consistency. In *3DV*, 2021. 3, 7, 9, 10

[12] Shuai Chen, Xinghui Li, Zirui Wang, and Victor Prisacariu. DFNet: Enhance absolute pose regression with direct feature matching. In *ECCV*, 2022. 1, 3, 7, 9

[13] Ronald Clark, Sen Wang, Andrew Markham, Niki Trigoni, and Hongkai Wen. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In *CVPR*, 2017. 3

[14] Siyan Dong, Shuzhe Wang, Yixin Zhuang, Juho Kannala, Marc Pollefeys, and Baoquan Chen. Visual localization via few-shot scene region classification. In *3DV*, 2022. 1, 2

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4

[16] Sovann En, Alexis Lechervy, and Frédéric Jurie. Rpnet: An end-to-end network for relative camera pose estimation. In *ECCVW*, 2018. 1

[17] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *CACM*, 1981. 1, 2

[18] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE TPAMI*, 2003. 1, 2

[19] Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time rgb-d camera relocalization. In *ISMAR*, 2013. 6, 9

[20] Google. ARCore. Accessed: 26 March 2024. 7

[21] 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. 1, 2

[22] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, 2020. 5

[23] A. Kendall and R. Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *ICRA*, 2016. 3, 7

[24] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017. 1, 3, 7, 9

[25] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015. 1, 3, 7, 8, 9

[26] Jake Levinson, Carlos Esteves, Kefan Chen, Noah Snavely, Angjoo Kanazawa, Afshin Rostamizadeh, and Ameesh Makadia. An analysis of svd for deep rotation estimation. *NeurIPS*, 2020. 1, 9

[27] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *CVPR*, pages 11983–11992, 2020. 2

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2017. 6

[29] Simon Lynen, Bernhard Zeisl, Dror Aiger, Michael Bosse, Joel Hesch, Marc Pollefeys, Roland Siegwart, and Torsten Sattler. Large-scale, real-time visual-inertial localization revisited. *International Journal of Robotics Research*, 39(9), 2019. 2

[30] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu. Image-based localization using hourglass networks. In *ICCVW*, 2017. 3, 7

[31] 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. 5

[32] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. LENS: Localization enhanced by nerf synthesis. In *CoRL*, 2021. 1, 3, 6, 7

[33] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Coordinet: uncertainty-aware pose regressor for reliable vehicle localization. In *WACV*, 2022. 3

[34] T. Naseer and W. Burgard. Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In *IROS*, 2017. 3

[35] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. Meshloc: Mesh-based visual localization. In *ECCV*, pages 589–609. Springer, 2022. 2

[36] Pulak Purkait, Cheng Zhao, and Christopher Zach. Synthetic view generation for absolute pose regression and image synthesis. In *BMVC*, 2018. 3

[37] N. Radwan, A. Valada, and W. Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. In *IEEE Robotics and Automation Letters*, 2018. 3

[38] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 1, 2

[39] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020.

[40] Paul-Edouard Sarlin, Ajaykumar Unagar, Måns Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, and Torsten Sattler. Back to the Feature: Learning robust camera localization from pixels to pose. In *CVPR*, 2021. 1, 2

[41] T. Sattler, B. Leibe, and L. Kobbelt. Improving image-based localization by active correspondence search. In *ECCV*, 2012. 2

[42] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. In *IEEE TPAMI*, 2017. 1, 2

[43] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *CVPR*, 2019. 1

[44] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 6

[45] Yoli Shavit and Yosi Keller. Camera pose auto-encoders for improving pose regression. In *ECCV*, 2022. 3

[46] Yoli Shavit, Ron Ferens, and Yosi Keller. Paying attention to activation maps in camera pose regression. In *arXiv preprint arXiv:2103.11477*, 2021. 1

[47] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *ICCV*, 2021. 3, 7, 8, 9

[48] 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. 6, 9

[49] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, 2019. 6

[50] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021. 4, 6

[51] Mehmet Özgür Türkoğlu, Eric Brachmann, Konrad Schindler, Gabriel Brostow, and Áron Monszpart. Visual Camera Re-Localization Using Graph Neural Networks and Relative Pose Supervision. In *3DV*, 2021. 1

[52] Julien Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *3DV*. IEEE, 2016. 10

[53] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-based localization using lstms for structured feature correlation. In *ICCV*, 2017. 3, 7

[54] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. In *AAAI*, 2020. 3

[55] Dominik Winkelbauer, Maximilian Denninger, and Rudolph Triebel. Learning to localize in new environments from synthetic training data. In *ICRA*, 2021. 1

[56] J. Wu, L. Ma, and X. Hu. Delving Deeper into Convolutional Neural Networks for Camera Relocalization. In *ICRA*, 2017. 3, 7

[57] Luwei Yang, Ziqian Bai, Chengzhou Tang, Honghua Li, Yasutaka Furukawa, and Ping Tan. SANet: Scene agnostic network for camera localization. In *ICCV*, 2019. 1, 2

[58] Qunjie Zhou, Sérgio Agostinho, Aljoša Ošep, and Laura Leal-Taixé. Is geometry enough for matching in visual localization? In *ECCV*, 2022. 1

[59] Yi Zhou, Connelly Barnes, Lu Jingwan, Yang Jimei, and Li Hao. On the continuity of rotation representations in neural networks. In *CVPR*, 2019. 5, 9

Statement of Authorship for the paper "Map-Relative Pose Regression for Visual Re-Localization" in Chapter 8.

Paper title	Map-Relative Pose Regression for Visual Re-Localization
Authors	**Shuai Chen**, Tommaso Cavallari, Victor Adrian Prisacariu, Eric Brachmann
Publication status	Published
Publication details	Conference on Computer Vision and Pattern Recognition (CVPR), 2024.

Student Confirmation

Student name	Shuai Chen	
Contribution to the paper	First-author contribution: • Conception of research ideas • Design and implementation of models • Running of large-scale experiments • Writing and presentation of the paper	
Signature and Date		Apr. 23th 2025

Supervisor Confirmation

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

Supervisor name	Prof. Victor Adrian Prisacariu	
Supervisor comments	The description is accurate	
Signature and Date		Apr. 23th 2025

Part IV

Conclusion

Chapter 9

Conclusion

This thesis has explored the integration of geometric priors into the Absolute Pose Regression paradigm to improve training, inference, and network architecture. While APR offers an efficient alternative to traditional structure-based visual relocalisation methods, its reliance on purely data-driven learning often limits its generalisation and accuracy. By incorporating geometric principles into the core of APR models, this research aims to bridge the gap between learning approaches and geometry-based techniques.

The first part of the thesis investigates how geometry awareness can be leveraged during training. Unlike conventional APR methods, which primarily rely solely on direct image-to-pose mappings, this thesis has proposed Direct-PoseNet and DFNet, two frameworks that integrate implicit 3D networks with the APR training process to provide photometric and feature-metric supervision. These approaches improve the generalisation and robustness of the APR in indoor and outdoor environments while enabling a more effective use of unlabelled data.

The second part focusses on inference-time enhancements, where existing APR methods typically lack mechanisms to refine predictions dynamically. The thesis introduces NeFeS and HR-APR to incorporate 3D geometric priors and uncertainty

estimation into APR frameworks. NeFeS refines APR predictions using a learned neural feature field, while HR-APR prioritises refinement based on uncertainty estimation to reduce computational overhead. Additionally, GS-CPR, a novel approach integrating 3D Gaussian splatting and the 3D foundation model MASt3R, provides a more efficient and scalable pose refinement framework. The proposed approach achieves state-of-the-art performance across several benchmarks.

The final part of the thesis examines how geometric priors can be embedded directly into the APR network architecture. A map-relative pose regression framework was developed by leveraging scene-specific geometric maps and a transformer-based architecture. This design enables the model to generalise across multiple environments while maintaining adaptability to new scenes with minimal fine-tuning. Unlike traditional APR methods that require extensive scene-specific training, this framework offers a more scalable approach to visual relocalisation, making APR more practical for real-world applications.

Overall, this thesis demonstrates the benefits of integrating geometric priors into APR across different stages of the pipeline. The methods presented contribute towards more robust and generalisable camera relocalisation approaches while maintaining the efficiency of learning-based techniques.

9.1 Future Work

While this thesis has addressed several fundamental challenges in APR-based camera relocalisation research, multiple open questions are worth further investigation. We highlight several promising research directions to extend our work, particularly for improving visual relocalisation algorithms.

An interesting future work direction is to develop hybrid localisation frameworks that bridge the gap between APR and geometry-based methods. As demon-

strated in this thesis, APR offers efficient inference and scalability, whereas geometry-based approaches provide strong interpretability and robustness in structured scenes. However, current APR and geometry-based models often operate in isolation from each other. Future work could investigate adaptive mechanisms that dynamically balance these two paradigms based on scene characteristics, environmental conditions, or resource constraints. For instance, one could explore applying APR-based inference in perceptually degraded environments (*e.g.,* under motion blur or fog) while reverting to geometric refinement when sufficient structure is available. The design of such flexible and context-aware strategies remains an open and valuable research challenge.

Another direction is to enhance the robustness of APR algorithms in more challenging scenarios, including severe appearance changes (*e.g.,* day-to-night transitions), seasonal variations, and large-scale urban environments. In these circumstances, APR has yet to surpass the adaptability of traditional geometry-based systems. Moreover, training 3D models such as NeRF and 3D-GS under such conditions remains a challenge.

Furthermore, multi-modal-based approaches are another frontier. Although this thesis primarily focuses on visual-only relocalisation, future research could investigate the fusion of visual, inertial, LiDAR point clouds, language embeddings, and semantic cues. Hypothetically speaking, these complementary inputs could compensate for the limitations of individual modalities. For example, inertial signals can improve the robustness of the algorithm under motion blur, while semantic understanding can disambiguate repetitive structures. By fusing diverse inputs, models may develop a more holistic understanding of the environment. This would be particularly valuable for applications in AR/VR and robotics. Future research could explore cross-modal contrastive learning, unified embedding spaces, or multi-sensor transformers tailored for this purpose.

From a computational efficiency standpoint, reducing the mapping and inference cost for SOTA accuracy APR methods remains a central challenge. This thesis introduced GS-CPR, which achieves a favourable trade-off between accuracy and runtime efficiency compared to other NeRF-based or GS-based refinement methods. However, constructing high-quality 3D-GS models suitable for visual relocalisation tasks still largely relies on dense multi-view imagery and time-consuming SfM point clouds. Further simplifying these requirements, especially under sparse or unconstrained conditions, could be beneficial.

In parallel, the utilisation of large-scale training datasets in obtaining emergent abilities for APR networks remains under-investigated. While *marepo* provides a practical approach to partially offload mapping computations through scene-independent pre-training, broader studies around scaling APR training to tens of millions of images, similar to recent vision-language and 3D foundation model trends, remain unexplored. Bridging this with APR, particularly through the integration of 3D foundation models [176, 88, 191, 42, 181, 103, 175], may offer a path toward more generalisable and efficient relocalisation systems.

Bibliography

[1] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[2] Relja Arandjelović and Andrew Zisserman. DisLocation: Scalable descriptor distinctiveness for location recognition. In *Proceedings of the Asian Conference on Computer Vision*, 2014.

[3] Daniel Barath, Dmytro Mishkin, Luca Cavalli, Paul-Edouard Sarlin, Petr Hruby, and Marc Pollefeys. Affineglue: Joint matching and robust estimation. *arXiv preprint arXiv:2307.15381*, 2023.

[4] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5855–5864, 2021.

[5] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022.

[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision*, 2006.

[7] Yash Bhalgat, Iro Laina, João F Henriques, Andrew Zisserman, and Andrea Vedaldi. Contrastive lift: 3d object instance segmentation by slow-fast contrastive fusion. *Advances in Neural Information Processing Systems*, 2023.

[8] Jia-Wang Bian, Wenjing Bian, Victor Adrian Prisacariu, and Philip Torr. Porf: Pose residual field for accurate neural surface reconstruction. In *Proceedings of the International Conference on Learning Representations*, 2024.

[9] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. NoPe-NeRF: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[10] Hunter Blanton, Connor Greenwell, Scott Workman, and Nathan Jacobs. Extending absolute pose regression to multiple scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020.

[11] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[12] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC - Differentiable RANSAC for Camera Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[13] Eric Brachmann and Carsten Rother. Neural-Guided RANSAC: Learning where to sample model hypotheses. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[14] Eric Brachmann and Carsten Rother. Visual camera re-localization from RGB and RGB-D images using DSAC. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[15] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-Aware Learning of Maps for Camera Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[16] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. *ACM Transactions on Graphics*, 2001.

[17] Huy-Hoang Bui, Bach-Thuan Bui, Dinh-Tuan Tran, and Joo-Ho Lee. Leveraging neural radiance field in descriptor synthesis for keypoints scene coordinate regression. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024.

[18] Mai Bui, Tolga Birdal, Haowen Deng, Shadi Albarqouni, Leonidas Guibas, Slobodan Ilic, and Nassir Navab. 6d camera relocalization in ambiguous scenes via continuous multimodal inference. In *Proceedings of the European Conference on Computer Vision*, pages 139–157. Springer, 2020.

[19] Federico Camposeco, Andrea Cohen, Marc Pollefeys, and Torsten Sattler. Hybrid scene compression for visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[20] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, 2020.

[21] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[22] Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM transactions on graphics (TOG)*, 32(3):1–12, 2013.

[23] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial radiance fields. In *Proceedings of the European Conference on Computer Vision*, 2022.

[24] Changhao Chen, Bing Wang, Chris Xiaoxuan Lu, Niki Trigoni, and Andrew Markham. Deep learning for visual localization and mapping: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[25] Guikun Chen and Wenguan Wang. A survey on 3d gaussian splatting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[26] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David Mckinnon, Yanghai Tsin, and Long Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In *Proceedings of the European Conference on Computer Vision*, pages 20–36. Springer, 2022.

[27] Le Chen, Weirong Chen, Rui Wang, and Marc Pollefeys. Leveraging neural radiance fields for uncertainty-aware visual localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 6298–6305. IEEE, 2024.

[28] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. *ACM Transactions on Graphics*, 2024.

[29] Shuai Chen, Yash Bhalgat, Xinghui Li, Jia-Wang Bian, Kejie Li, Zirui Wang, and Victor Adrian Prisacariu. Neural refinement for absolute pose regression with fea-

ture synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20987–20996, 2024.

[30] Shuai Chen, Xinghui Li, Zirui Wang, and Victor Prisacariu. DFNet: Enhance absolute pose regression with direct feature matching. In *Proceedings of the European Conference on Computer Vision*, 2022.

[31] Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-posenet: Absolute pose regression with photometric consistency. In *3DV*. IEEE, 2021.

[32] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16569–16578, 2023.

[33] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[34] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[35] Yuzhou Cheng, Jianhao Jiao, Yue Wang, and Dimitrios Kanoulas. Logs: Visual localization via gaussian splatting with fewer training images. *arXiv preprint arXiv:2410.11505*, 2024.

[36] Ronald Clark, Sen Wang, Andrew Markham, Niki Trigoni, and Hongkai Wen. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[37] Anurag Dalal, Daniel Hagen, Kjell G Robbersmyr, and Kristian Muri Knausgård. Gaussian splatting: 3d reconstruction and novel view synthesis, a review. *IEEE Access*, 2024.

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

[39] Frank Dellaert and Lin Yen-Chen. Neural volume rendering: Nerf and beyond. *arXiv preprint arXiv:2101.05204*, 2020.

[40] Haowen Deng, Mai Bui, Nassir Navab, Leonidas Guibas, Slobodan Ilic, and Tolga Birdal. Deep bingham networks: Dealing with uncertainty and ambiguity in pose estimation. *International Journal of Computer Vision*, 130(7):1627–1654, 2022.

[41] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[42] Siyan Dong, Shuzhe Wang, Shaohui Liu, Lulu Cai, Qingnan Fan, Juho Kannala, and Yanchao Yang. Reloc3r: Large-scale training of relative camera pose regression for generalizable, fast, and accurate visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[43] Siyan Dong, Shuzhe Wang, Yixin Zhuang, Juho Kannala, Marc Pollefeys, and Baoquan Chen. Visual localization via few-shot scene region classification. In *3DV*, 2022.

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

[45] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A trainable cnn for joint detection and description of local features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[46] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. DKM: Dense kernelized feature matching for geometry estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17765–17775, 2023.

[47] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust dense feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19790–19800, 2024.

[48] Martin Eisemann, Bert De Decker, Marcus Magnor, Philippe Bekaert, Edilson De Aguiar, Naveed Ahmed, Christian Theobalt, and Anita Sellent. Floating textures. In *Computer graphics forum*. Wiley Online Library, 2008.

[49] Jakob Engel, Vladlen Koltun, and Daniel Cremers. DSO: Direct sparse odometry. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[50] Jakob Engel, Jurgen Sturm, and Daniel Cremers. Semi-dense visual odometry for a monocular camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2013.

[51] Ben Fei, Jingyi Xu, Rui Zhang, Qingyuan Zhou, Weidong Yang, and Ying He. 3d gaussian splatting as new era: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[52] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, 1981.

[53] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[54] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and Xiaolong Wang. COLMAP-Free 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20796–20805, June 2024.

[55] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2009.

[56] 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.

[57] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.

[58] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14346–14355, 2021.

[59] Marcel Geppert, Peidong Liu, Zhaopeng Cui, Marc Pollefeys, and Torsten Sattler. Efficient 2d-3d matching for multi-camera visual localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5972–5978. IEEE, 2019.

[60] Hugo Germain, Daniel DeTone, Geoffrey Pascoe, Tanner Schmidt, David Novotny, Richard Newcombe, Chris Sweeney, Richard Szeliski, and Vasileios Balntas. Feature query networks: Neural surface description for camera pose refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5071–5081, 2022.

[61] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M. Seitz. Multi-view stereo for community photo collections. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–8, 2007.

[62] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. Deep image retrieval: Learning global representations for image search. In *Proceedings of the European Conference on Computer Vision*, 2016.

[63] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996.

[64] Seongbo Ha, Jiung Yeon, and Hyeonwoo Yu. Rgbd gs-icp slam. In *Proceedings of the European Conference on Computer Vision*, pages 180–197. Springer, 2024.

[65] Juyeop Han, Lukas Lao Beyer, Guilherme V Cavalheiro, and Sertac Karaman. NVINS: Robust visual inertial navigation fused with nerf-augmented camera pose regressor and uncertainty quantification. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024.

[66] Richard Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997.

[67] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (ToG)*, 37(6):1–15, 2018.

[68] Jiarui Hu, Xianhao Chen, Boyin Feng, Guanglin Li, Liangjing Yang, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. CG-SLAM: Efficient dense rgb-d slam in a consistent uncertainty-aware 3d gaussian field. In *Proceedings of the European Conference on Computer Vision*, pages 93–112. Springer, 2024.

[69] Zhaoyang Huang, Yan Xu, Jianping Shi, Xiaowei Zhou, Hujun Bao, and Guofeng Zhang. Prior guided dropout for robust visual localization in dynamic environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2791–2800, 2019.

[70] Zhiwei Huang, Hailin Yu, Yichun Shentu, Jin Yuan, and Guofeng Zhang. From sparse to dense: Camera relocalization with scene-specific detector from feature gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[71] 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.

[72] Michal Irani and Prabu Anandan. All about direct methods. In *International Workshop on Vision Algorithms*, 1999.

[73] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2599–2606. IEEE, 2009.

[74] Peng Jiang, Gaurav Pandey, and Srikanth Saripalli. 3DGS-ReLoc: 3d gaussian splatting for map representation and visual relocalization. *arXiv preprint arXiv:2403.11367*, 2024.

[75] Gyeongjin Kang, Jisang Yoo, Jihyeon Park, Seungtae Nam, Hyeonsoo Im, Sangpil Kim, Eunbyung Park, et al. Selfsplat: Pose-free and 3d prior-free generalizable 3d gaussian splatting. *arXiv preprint arXiv:2411.17190*, 2024.

[76] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. SplaTAM: Splat Track & Map 3D Gaussians for Dense RGB-D SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21357–21366, 2024.

[77] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2016.

[78] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[79] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2015.

[80] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 2023.

[81] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. In *Advances in Neural Information Processing Systems*, 2022.

[82] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. In *Computer Graphics Forum*, 2021.

[83] Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.

[84] Christoph Lassner and Michael Zollhofer. Pulsar: Efficient sphere-based neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[85] JongMin Lee and Sungjoo Yoo. Dense-SfM: Structure from motion with dense consistent matching. *arXiv preprint arXiv:2501.14277*, 2025.

[86] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[87] Miso Lee, Jihwan Kim, and Jae-Pil Heo. Activating self-attention for multi-scene absolute pose regression. In *Advances in Neural Information Processing Systems*, 2024.

[88] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding Image Matching in 3D with MASt3R. In *Proceedings of the European Conference on Computer Vision*, pages 71–91. Springer, 2024.

[89] Jake Levinson, Carlos Esteves, Kefan Chen, Noah Snavely, Angjoo Kanazawa, Afshin Rostamizadeh, and Ameesh Makadia. An Analysis of SVD for Deep Rotation Estimation. *Advances in Neural Information Processing Systems*, 2020.

[90] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996.

[91] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *Proceedings of the International Conference on Learning Representations*, 2022.

[92] Chunming Li, Chenyang Xu, Changfeng Gui, and Martin D Fox. Distance regularized level set evolution and its application to image segmentation. *IEEE Transactions on Image Processing*, 2010.

[93] Hao Li, Yuanyuan Gao, Dingwen Zhang, Chenming Wu, Yalun Dai, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Junwei Han. GGRt: Towards generalizable 3d gaussians without pose priors in real-time. *arXiv preprint arXiv:2403.10147*, 2024.

[94] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11983–11992, 2020.

[95] Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-resolution correspondence networks. In *Advances in Neural Information Processing Systems*, 2020.

[96] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[97] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[98] Jingyu Lin, Jiaqi Gu, Bojian Wu, Lubin Fan, Renjie Chen, Ligang Liu, and Jieping Ye. Learning neural volumetric pose features for camera localization. In *Proceedings of the European Conference on Computer Vision*, 2024.

[99] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17627–17638, 2023.

[100] Jianlin Liu, Qiang Nie, Yong Liu, and Chengjie Wang. Nerf-loc: Visual localization with conditional neural radiance field. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 9385–9392. IEEE, 2023.

[101] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 2020.

[102] Liu Liu, Hongdong Li, and Yuchao Dai. Efficient global 2d-3d matching for camera localization in a large-scale 3d map. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2372–2381, 2017.

[103] Yuzheng Liu, Siyan Dong, Shuzhe Wang, Yanchao Yang, Qingnan Fan, and Baoquan Chen. Slam3r: Real-time dense scene reconstruction from monocular rgb videos. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[104] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two images. *Nature*, 1981.

[105] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[106] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-GS: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[107] Simon Lynen, Bernhard Zeisl, Dror Aiger, Michael Bosse, Joel Hesch, Marc Pollefeys, Roland Siegwart, and Torsten Sattler. Large-scale, real-time visual-inertial localization revisited. *International Journal of Robotics Research*, 39(9), 2019.

[108] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[109] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Image-based localization using hourglass networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2017.

[110] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[111] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proceedings of the European Conference on Computer Vision*, 2020.

[112] Saad Mokssit, Daniel Bonilla Licea, Bassma Guermah, and Mounir Ghogho. Deep learning techniques for visual slam: A survey. *IEEE Access*, 11:20026–20050, 2023.

[113] Arthur Moreau, Nathan Piasco, Moussab Bennehar, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Crossfire: Camera relocalization on self-supervised features from an implicit representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 252–262, 2023.

[114] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. LENS: Localization enhanced by nerf synthesis. In *CoRL*, 2021.

[115] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. CoordiNet: uncertainty-aware pose regressor for reliable vehicle localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022.

[116] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 2022.

[117] Tayyab Naseer and Wolfram Burgard. Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

[118] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H Mueller, Chakravarty R Alla Chaitanya, Anton Kaplanyan, and Markus Steinberger. DONeRF: Towards real-time rendering of compact neural radiance fields using depth oracle networks. In *Computer Graphics Forum*, 2021.

[119] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2011.

[120] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021.

[121] Zhongyan Niu, Zhen Tan, Jinpu Zhang, Xueliang Yang, and Dewen Hu. HGSLoc: 3dgs-based heuristic camera pose refinement. *arXiv preprint arXiv:2409.10925*, 2024.

[122] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion*. *Acta Numerica*, 26:305–364, 2017.

[123] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. Meshloc: Mesh-based visual localization. In *Proceedings of the European Conference on Computer Vision*, pages 589–609. Springer, 2022.

[124] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[125] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021.

[126] Zhexi Peng, Tianjia Shao, Yong Liu, Jingke Zhou, Yin Yang, Jingdong Wang, and Kun Zhou. Rtg-slam: Real-time 3d reconstruction at scale using gaussian splatting. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[127] Nathan Piasco, Désiré Sidibé, Cédric Demonceaux, and Valérie Gouet-Brunet. A survey on visual-based localization: On the benefit of heterogeneous data. *Pattern Recognition*, 74:90–109, 2018.

[128] Matia Pizzoli, Christian Forster, and Davide Scaramuzza. Remode: Probabilistic, monocular dense reconstruction in real time. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2014.

[129] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[130] Pulak Purkait, Cheng Zhao, and Christopher Zach. Synthetic view generation for absolute pose regression and image synthesis. In *BMVC*, 2018.

[131] Noha Radwan, Abhinav Valada, and Wolfram Burgard. VLocNet++: Deep multitask learning for semantic visual localization and odometry. In *IEEE Robotics and Automation Letters*, 2018.

[132] Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. Derf: Decomposed radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14153–14161, 2021.

[133] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14335–14345, 2021.

[134] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. In *Advances in Neural Information Processing Systems*, 2019.

[135] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proceedings of the European Conference on Computer Vision*, pages 430–443. Springer, 2006.

[136] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[137] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[138] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[139] Paul-Edouard Sarlin, Ajaykumar Unagar, Måns Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, and Torsten Sattler. Back to the Feature: Learning robust camera localization from pixels to pose. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[140] Torsten Sattler, Michal Havlena, Konrad Schindler, and Marc Pollefeys. Large-scale location recognition and the geometric burstiness problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[141] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 667–674. IEEE, 2011.

[142] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *Proceedings of the European Conference on Computer Vision*, 2012.

[143] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[144] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[145] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[146] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision*, pages 501–518. Springer, 2016.

[147] David Schubert, Nikolaus Demmel, Vladyslav Usenko, Jorg Stuckler, and Daniel Cremers. Direct sparse odometry with rolling shutter. In *Proceedings of the European Conference on Computer Vision*, pages 682–697, 2018.

[148] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *Advances in Neural Information Processing Systems*, 2020.

[149] Steven M. Seitz and Charles R. Dyer. View morphing. *ACM Transactions on Graphics*, 1996.

[150] Steven M. Seitz and Charles R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999.

[151] Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. In *Workshop on Language and Robotics at CoRL*, 2022.

[152] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[153] Yoli Shavit, Ron Ferens, and Yosi Keller. Paying attention to activation maps in camera pose regression. In *arXiv preprint arXiv:2103.11477*, 2021.

[154] Yoli Shavit and Yosi Keller. Camera pose auto-encoders for improving pose regression. In *Proceedings of the European Conference on Computer Vision*, 2022.

[155] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2013.

[156] Harry Shum and Sing Bing Kang. Review of image-based rendering techniques. *Visual Communications and Image Processing 2000*, 4067:2–13, 2000.

[157] Yoli Shvit and Ron Ferens. Introduction to camera pose estimation with deep learning. *arXiv preprint arXiv: 1907.05272*, 2019.

[158] Gennady Sidorov, Malik Mohrat, Ksenia Lebedeva, Ruslan Rakhimov, and Sergey Kolyubin. GSplatLoc: Grounding keypoint descriptors into 3d gaussian splatting for improved visual localization. *arXiv preprint arXiv:2409.16502*, 2024.

[159] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structureaware neural scene representations. In *Advances in Neural Information Processing Systems*, 2019.

[160] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3d. In *ACM siggraph*, pages 835–846. ACM, 2006.

[161] Jan Stühmer, Stefan Gumhold, and Daniel Cremers. Real-time dense geometry from a handheld camera. In *Joint Pattern Recognition Symposium*, pages 11–20. Springer, 2010.

[162] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew Davison. iMAP: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[163] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022.

[164] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[165] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, , and Akihiko Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[166] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. State of the art on neural rendering. In *Computer Graphics Forum*, 2020.

[167] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 1992.

[168] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2011.

[169] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural Feature Fusion Fields: 3D distillation of self-supervised 2D image representations. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2022.

[170] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *Advances in Neural Information Processing Systems*, 33:14254–14265, 2020.

[171] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[172] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

[173] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. AtLoc: Attention guided camera localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[174] Fangjinhua Wang, Xudong Jiang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. Glace: Global local accelerated coordinate encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21562–21571, 2024.

[175] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[176] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

[177] 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.

[178] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet - photo geolocation with convolutional neural networks. In *Proceedings of the European Conference on Computer Vision*, 2016.

[179] Jian Wu, Liwei Ma, and Xiaolin Hu. Delving Deeper into Convolutional Neural Networks for Camera Relocalization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2017.

[180] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9421–9431, 2021.

[181] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3R: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[182] Luwei Yang, Ziqian Bai, Chengzhou Tang, Honghua Li, Yasutaka Furukawa, and Ping Tan. SANet: Scene agnostic network for camera localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[183] 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 *arxiv arXiv:2012.05877*, 2020.

[184] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *Proceedings of the European Conference on Computer Vision*, pages 467–483. Springer, 2016.

[185] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[186] Amir Roshan Zamir and Mubarak Shah. Accurate image localization based on google maps street view. In *Proceedings of the European Conference on Computer Vision*, 2010.

[187] Fereidoon Zangeneh, Leonard Bruns, Amit Dekel, Alessandro Pieropan, and Patric Jensfelt. A probabilistic framework for visual localization in ambiguous scenes. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3969–3975, 2023.

[188] Atticus J Zeller. Gsplatloc: Ultra-precise camera localization via 3d gaussian splatting. *arXiv preprint arXiv:2412.20056*, 2024.

[189] Hongjia Zhai, Xiyu Zhang, Boming Zhao, Hai Li, Yijia He, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. Splatloc: 3d gaussian splatting-based visual localization for augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 2025.

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

[191] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[192] Boming Zhao, Luwei Yang, Mao Mao, Hujun Bao, and Zhaopeng Cui. PNeRFLoc: Visual localization with point-based neural radiance fields. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[193] Qunjie Zhou, Maxim Maximov, Or Litany, and Laura Leal-Taixé. The nerfect match: Exploring nerf features for visual localization. In *Proceedings of the European Conference on Computer Vision*, pages 108–127. Springer, 2024.

[194] Yi Zhou, Connelly Barnes, Lu Jingwan, Yang Jimei, and Li Hao. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[195] Zhiyu Zhou, Feng Hui, Yilin Wu, and Yu Liu. Six-DoF pose estimation with efficient 3-d gaussian splatting representation for visual relocalization. *IEEE/ASME Transactions on Mechatronics*, 2024.

[196] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. NICE-SLAM: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2022.

[197] C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM transactions on graphics (TOG)*, 23(3):600–608, 2004.