GRLoc: Geometric Representation Regression for Visual Localization

Changyang Li Xuejian Ma Lixiang Liu Zhan Li Qingan Yan Yi Xu

Goertek Alpha Labs
{first.last}@goertekusa.com

Abstract. Absolute Pose Regression (APR) has emerged as a compelling paradigm for visual localization. However, APR models typically operate as black boxes, directly regressing a 6-DoF pose from a query image, which can lead to memorizing training views rather than understanding 3D scene geometry. In this work, we propose a geometrically-grounded alternative. Inspired by novel view synthesis, which renders images from intermediate geometric representations, we reformulate APR as its inverse that regresses the underlying 3D representations directly from the image, and we name this paradigm Geometric Representation Regression (GRR). Our model explicitly predicts two disentangled geometric representations in the world coordinate system: (1) a raymap's directions to estimate camera rotation, and (2) a corresponding pointmap to estimate camera translation. The final camera pose is then recovered from these geometric components using a differentiable deterministic solver. This disentangled approach, which separates the learned visual-to-geometry mapping from the final pose calculation, introduces a strong geometric prior into the network. We find that the explicit decoupling of rotation and translation predictions measurably boosts performance. We demonstrate state-of-the-art performance on 7-Scenes and Cambridge Landmarks datasets, validating that modeling the inverse rendering process is a more robust path toward generalizable absolute pose estimation.

Keywords: visual localization, camera pose estimation.

1 Introduction

The paradigm of Absolute Pose Regression (APR), where a single neural network learns to predict a camera's 6-DoF pose directly from a query image, holds immense promise for simple and efficient visual localization systems [9, 28, 48]. This end-to-end approach contrasts with traditional methods that rely on complex, multi-stage pipelines of feature matching and geometric verification. However, the apparent simplicity of APR has been challenged by a critical finding: these networks often fail to generalize to novel viewpoints, instead functioning as sophisticated memory banks that retrieve the poses of the most visually similar training images [47]. This tendency to "memorize" rather than "understand" suggests that the models are learning an opaque mapping from pixels to pose, bypassing the foundational geometric principles of camera projection.

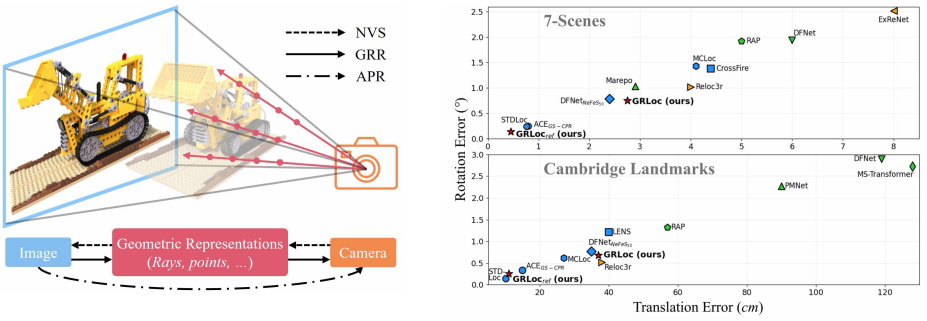

Fig. 1: Left: Our Geometric Representation Regression (GRR) paradigm is inspired by Novel View Synthesis (NVS). While NVS performs the forward rendering process, GRR learns the inverse. Right: Localization errors on the 7-Scenes [50] and Cambridge [28] datasets. Our GRLoc outperforms prior APR (green) methods and demonstrates competitive performance against leading RPR (orange) and PPR (blue) approaches. Our refined model $\text{GRLoc}_{\text{ref}}$ achieves comparable results with SOTA PPR methods.

This reveals a fundamental issue not in the network's capacity, but in how the task is formulated. Tasking a model with learning the direct, highly non-linear transformation from a 2D image to a 6-DoF pose vector is challenging because it encourages the network to discover statistical shortcuts. While synthetic data augmentation [12, 31, 32, 40] can expand the "memory bank", it does not fundamentally change the black-box nature of the learning process.

The evolution of novel view synthesis (NVS) offers an instructive parallel. Recent methods like Neural Radiance Field (NeRF) [38] and 3D Gaussian Splatting (3DGS) [29] decompose rendering into explicit geometric steps. Given camera parameters, these methods cast **rays** into the scene, sample **points** along rays or query Gaussians, accumulate appearance, and render the final image. This explicit, geometry-centric decomposition has proven far more effective at generating high-fidelity views. This success has inspired methods to invert the process, such as iNeRF [59], which iteratively refines a pose via analysis-by-synthesis.

In this paper, we propose a paradigm shift that applies this principle in reverse for pose estimation. Just as NeRF and 3DGS decompose the forward rendering process, we decompose the inverse problem into a single-pass regression task, fundamentally different from iNeRF [59]: given an image, we first regress its underlying geometric representations, and then compute the camera pose from this structure. This idea is illustrated in Figure 1. Specifically, our network predicts two geometric representations for each image patch: a **raymap**'s directions to estimate camera rotation, and a **pointmap** to estimate camera translation.

We name this paradigm Geometric Representation Regression (GRR), and name our visual localization approach GRLoc. While our method's internal structure is different, it maintains the end-to-end simplicity of APR: it takes an image as input and produces a 6-DoF pose as output. The camera pose is then deterministically recovered from these geometric representations using a

differentiable solver. Since the forward rendering in NeRF/3DGS is mathematically uninvertible in principle, this practical inversion requires learned perception to estimate geometric representations from image context. We leverage 3DGS to synthesize diverse training views, expanding scene coverage without requiring further real-world data collection.

A critical finding of this work is that decoupling the predictions for rotation and translation is key to improving performance. While balancing the optimization of both targets in a single branch is theoretically feasible, it is practically challenging. Our model design thus splits the network into a ray branch for rotation prediction and a point branch for translation prediction.

This hybrid framework separates learned perception (image to geometry) from analytical computation (geometry to pose), introducing a strong geometric prior and improving interpretability. Through domain-adversarial training to align synthetic and real data, our method achieves robust generalization. Our main contributions are:

1. We reformulate absolute pose estimation as an inverse rendering task, decomposing the problem into a learned perception step (image to geometry) and an analytical solving step (geometry to pose).
2. We introduce GRLoc that predicts 3D rays and points, and then converts them to a pose via a differentiable solver. GRLoc achieves state-of-the-art performance on the 7-Scenes and Cambridge Landmarks datasets.
3. We find that rotation and translation are competing objectives, and demonstrate that our decoupled geometric representations and network designs are critical for avoiding this conflict and achieving high accuracy.

2 Related Work

2.1 Visual Localization Paradigms

Camera pose estimation is a long-standing problem in computer vision. A well-studied category of geometry-based methods establish correspondences between 2D keypoints in a query image and a sparse point cloud [18, 23, 25, 33, 36, 42–46, 52, 60], and use these 2D-3D matches to solve for the 6-DoF pose via PnP [21] algorithms within a RANSAC [19] loop. These methods achieve high accuracy due to their explicit geometric reasoning, but require storing and querying large 3D point clouds, raising concerns about storage and computation costs.

A more recent family of work, Scene Coordinate Regression (SCR) [5, 7, 8, 26, 53, 56], replaces the reliance on local feature matching by training a neural network to directly predict dense, pixel-aligned 3D world coordinates for the query image. SCR methods maintain the geometric robustness of correspondence-based approaches while using compact implicit scene representations. Notable works include ACE [5], which introduced efficient scene-centric training, and GLACE [56], which improved scalability to large scenes. However, SCR methods still face challenges, including the reliance on a time-consuming geometric solver step, typically RANSAC-PnP [19, 21].

Another paradigm is Relative Pose Regression (RPR) [1–3, 15, 16, 57], which regresses the relative pose between the query image and the most similar images in the scene database as references. RPR methods can leverage the rich information from reference images to improve pose accuracy, particularly in challenging scenarios. A main limitation is the image retrieval step needed for preparing the reference images, which can be time-consuming, especially in large scene databases, and the reliance on reference image quality.

In contrast, Absolute Pose Regression (APR) [4, 9, 11–13, 28, 31, 32, 48, 49] aims for a fully end-to-end solution, training a deep neural network to directly map an input image to a 6-DoF pose. The primary advantage of APR is its inference speed and simplicity, as it collapses a multi-stage pipeline into a single forward pass. However, this end-to-end solution has historically come at the cost of accuracy and interpretability, with APR models often achieving lower accuracy than geometry-based methods. The fundamental challenge lies in the compact nature of the pose representation: a 6-parameter vector is highly sensitive to noise and provides limited redundancy for optimization. This issue has been addressed from several perspectives, such as augmenting the model with larger-scale [12, 32, 40] or more diverse [31] training data.

Instead, we fundamentally reconsider the architecture of the regressor itself. We propose that the network's objective should not be to predict the final pose directly, but rather to infer intermediate geometric representations from which the pose can be recovered through well-established geometric algorithms. Our approach is inspired by the explicit reasoning of classical methods, but formulates the geometry prediction as a dense, learned task suitable for end-to-end training. Our method acts as an "inverse camera," deconstructing an image into its constituent 3D components: a raymap and a pointmap. This disentangled 'image $\rightarrow$ geometry $\rightarrow$ pose' framework contrasts with the 'image $\rightarrow$ pose' paradigm of APR, and the 'image $\rightarrow$ 3D coordinates $\rightarrow$ pose' pipeline of SCR, offering a new path toward building more robust and interpretable pose estimators.

Although the network's direct output differs from conventional APR, our GRR method remains fully end-to-end. The model's input (an image) and final output (a 6-DoF pose) are identical to APR, as the intermediate geometric representations are passed to a fully differentiable solver. Furthermore, the primary training objective aligns with APR: optimizing for the correctness of the final pose. Notably, while recent methods like Marepo [11] are recognized as APRs [34, 51] despite explicitly predicting scene coordinates (SCR style), GRLoc avoids dense coordinate maps by estimating camera-specific, patch-level geometry, making it conceptually closer to conventional APR.

A recent work CRR [61] also investigates ray regression. Our GRR differs in three key aspects: (1) We aim to "invert" rendering to learn generalizable 3D geometry, whereas CRR focuses on resolving the privacy and unbounded error issues of SCR. (2) CRR operates as a non-end-to-end SCR pipeline relying on RANSAC-PnP. (3) CRR utilizes Plücker coordinates, which entangle rotation and translation. Conversely, GRR explicitly decouples them following a design choice we found critical for optimal performance.

2.2 Novel View Synthesis for Localization

Recent advancements in novel view synthesis (NVS) have demonstrated impressive capabilities in reconstructing and rendering 3D scenes. Neural Radiance Fields (NeRF) [38] pioneered the use of neural implicit representations to encode scene geometry and appearance in the weights of a multilayer perceptron, enabling photorealistic rendering of novel viewpoints through volumetric ray marching. Building on this foundation, 3D Gaussian Splatting (3DGS) [29] introduced an explicit representation using 3D Gaussians, achieving real-time rendering speeds while maintaining high visual fidelity through efficient rasterization. These NVS methods have opened new opportunities for enhancing visual localization in multiple ways.

First, NVS provides a powerful data augmentation mechanism for expanding limited training datasets. APR methods have successfully leveraged NeRF [32,40] and 3DGS [12, 31] to synthesize large quantities of novel views with accurate ground-truth poses, addressing the data scarcity problem that has long hindered learning-based localization. This synthetic-to-real transfer has proven particularly valuable for training robust pose regressors, though it introduces the challenge of bridging the domain gap between rendered and real images.

Another family of work, Post-Pose Refinement (PPR) methods [22, 35, 54], aims to iteratively refine an initial coarse pose estimate. Many recent PPR methods also leverage NVS [10, 24, 34, 39, 41, 51, 62, 63]. These methods pass an initial pose prediction to an NVS model, render a synthetic view from that estimated viewpoint, and refine the predicted pose by minimizing the photometric or feature-based discrepancy between the rendered view and the query image. This "render and compare" strategy effectively converts the pose estimation problem into an optimization process, achieving impressive refinement accuracy when starting from reasonable initial estimates.

Our method also utilizes 3DGS, drawing inspiration from both approaches. First, similar to APR methods, we leverage 3DGS to generate synthetic training views that significantly expand our training data beyond the limited real-world images available for a given scene, allowing our model to learn more robust geometric representations by observing the scene from diverse viewpoints. Second, we incorporate 3DGS-based post-refinement following the settings of RAP [31] as an optional step to further improve the pose accuracy. However, our primary contribution lies in the initial pose estimation via GRR, not the refinement process itself. While we demonstrate that our method is compatible with post-refinement, we treat it as an optional step to fairly evaluate our core approach.

3 Method

We formalize the absolute pose estimation problem as follows. Given a single RGB image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, our goal is to estimate the camera pose represented by a camera-to-world transformation matrix $\mathbf{T} \in \mathrm{SE}(3)$, which includes a rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$ representing the camera center in world coordinates.

Rather than directly regressing the pose parameters, we reformulate this problem as estimating intermediate geometric representations that encode the 3D structure of the camera's view. Specifically, we decompose the image into a grid of $N \times N$ patches.

While the forward rendering process of NVS (casting rays to form pixels) is well-defined, its inverse is not. Our goal of "inverting" this process, which is to estimate 3D representations from a 2D image, is a learned perception task. It requires considering both local patch information and global spatial relationships, a task that aligns well with Vision Transformer [17] architectures. We predict two types of **geometric representations** for each patch i, as illustrated in Figure 2:

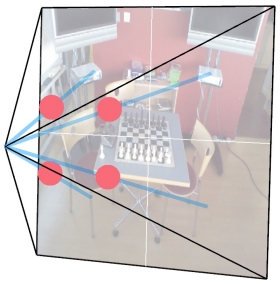

Fig. 2: Geometric representations for an example 4×4 grid: (1) a raymap of view directions and (2) a pointmap of 3D points.

- **Ray direction field** $\hat{\mathcal{D}} = \{\hat{\mathbf{d}}_i \in \mathbb{S}^2\}_{i=1}^{N \times N}$: A set of predicted unit vectors in world coordinates representing the viewing direction corresponding to patch i. Each patch in the grid acts as a 'super-pixel', and its predicted ray $\hat{\mathbf{d}}_i$ represents the average direction of all pixel-rays within that region. For simplicity, we refer to these as "rays", though they technically represent direction vectors.
- **3D pointmap** $\hat{\mathcal{P}} = \{\hat{\mathbf{p}}_i \in \mathbb{R}^3\}_{i=1}^{N \times N}$: A set of predicted points in world coordinates. Each point $\hat{\mathbf{p}}_i$ is modeled to lie on the corresponding ray $\hat{\mathbf{d}}_i$ and is constrained to be at a unit distance from the camera's origin ($\|\hat{\mathbf{p}}_i - \mathbf{t}\| = 1$), which removes scale ambiguity. This constraint defines the pointmap in a camera-relative canonical space (like directional vectors) and allows recovering translation without regressing unbounded world coordinates.

Given these predicted representations $\hat{\mathcal{D}}$ and $\hat{\mathcal{P}}$, and their known canonical (in camera space) counterparts $\mathcal{D}^{\text{cam}}$ and $\mathcal{P}^{\text{cam}}$, our goal is to recover the camera pose. As our ablation studies (Section 4.4) suggest, rotation and translation can be competing objectives when entangled. We therefore solve for them using two specialized, differentiable alignment problems. At a high level, this involves solving for the transformation $\mathbf{T}$ that minimizes the joint error:

$$\mathbf{T} = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{N^2} \|\mathbf{R}\mathbf{p}_i^{\text{cam}} - \mathbf{d}_i\|^2 + \|\mathbf{R}\mathbf{p}_i^{\text{cam}} + \mathbf{t} - \mathbf{p}_i\|^2 \tag{1}$$

where the first term formulates a rotation-only alignment problem for the rays, and the second term formulates a rigid 3D registration problem for the points.

We discretize the problem by predicting one representation per patch in an $N \times N$ grid. While a denser grid (i.e., 1 pixel per patch) could theoretically improve the solver's robustness by providing more constraints, this benefit is conditional on the network's ability to predict these dense representations accurately. Therefore, selecting the patch grid density requires a trade-off between geometric detail coverage and per-patch prediction quality.

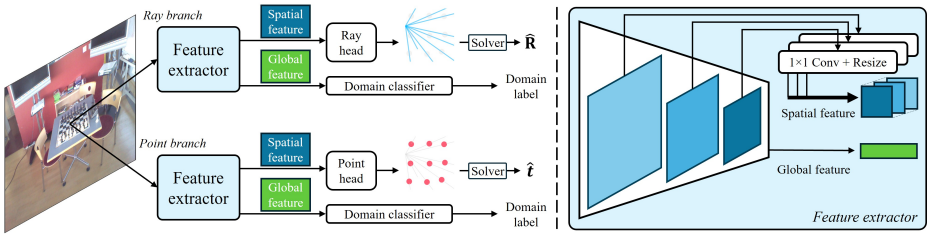

Fig. 3: An overview of our proposed architecture. Left: the query image is fed into a decoupled dual-branch network. Each branch's feature extractor produces spatial features for a prediction head and a global feature for a domain classifier. The **ray head** and **point head** predict their respective geometric representations. The final pose components are then analytically recovered from these representations using differentiable deterministic solvers. Right: The feature extractor fuses multi-level features from the backbone via 1×1 convolutions, resizing, and concatenation to produce the spatial feature, while the backbone's original final output is used as the global feature.

An overview of our architecture is shown in Figure 3. The input image is processed by a dual-branch network, which is intentionally decoupled to support this specialized alignment. The ray branch predicts per-patch ray directions, while the point branch predicts per-patch 3D points. These predictions are fed into their respective deterministic, differentiable solvers, allowing us to leverage their complementary strengths in the final loss: we use the rotation computed from the ray alignment and the translation computed from the point alignment.

3.1 Geometric Representation Prediction

Our model's architecture is intentionally designed as a decoupled dual-branch network to predict the two distinct geometric representations from a single RGB image. This design prevents the competing objectives of rotation and translation from interfering with each other at the feature level.

Feature Extractors. Each of the *ray branch* and the *point branch* has its own feature extractor built with a transformer-based backbone and a fusion neck. The extractor's job is to produce two outputs: a single global feature $\mathbf{f}_{global}$ for domain alignment, and a spatially-aware multi-level feature map $\mathbf{F}_{spatial}$ for the prediction heads.

A feature extractor uses a transformer-based backbone, which is well-suited for our per-patch prediction task of geometric representation prediction. To capture rich local information, each branch's multi-level feature maps are processed by a dedicated fusion neck, as shown in Figure 3 (Right). This neck unifies their channel dimensions via 1×1 convolutions, resizes them to a target spatial resolution of $N \times N$, and concatenates them. This produces a branch-specific, spatially-aware feature tensor $\mathbf{F}_{spatial}$, where each spatial location corresponds to an image patch.

Prediction Heads. Two lightweight MLP heads, *ray head* and *point head*, regress the geometric representations from their respective branch's spatial features $\mathbf{F}_{spatial}$, and output the dense ray direction field $\hat{\mathcal{D}}$ and 3D pointmap $\hat{\mathcal{P}}$.

3.2 Deterministic Pose Solver

A key aspect of our method is that the final camera pose is not directly predicted. Instead, we estimate it deterministically from the predicted geometric representations using non-learnable but fully differentiable solvers. This step offloads the complex, non-linear pose estimation from the network and allows for end-to-end training. As our design is decoupled, we compute two separate pose estimates to provide specialized supervision for rotation and translation.

Both of our alignment tasks (the rotation-only Procrustes problem for rays and the rigid point set registration for points) are solved using the Kabsch algorithm [27], which provides a closed-form solution for finding the optimal rigid transformation between two sets of corresponding vectors or points. At its core, the algorithm relies on a Singular Value Decomposition (SVD) of a covariance matrix. This makes the solver highly efficient and, crucially for our end-to-end framework, fully differentiable.

Rotation from Rays. The ray branch is specialized for rotation. We solve a rotation-only Procrustes problem to find the optimal rotation $\hat{\mathbf{R}} \in \mathrm{SO}(3)$ that best aligns the predicted world-space ray directions $\hat{\mathcal{D}}$ with the canonical ray directions $\mathcal{D}^{\mathrm{cam}}$. This approach is critical because the raymap is a pure representation of rotation, free from the competing objective of translation.

Translation from Points. We find the full transformation $[\hat{\mathbf{R}}_{\mathrm{point}}|\hat{\mathbf{t}}_{\mathrm{point}}]$ that best aligns the predicted 3D point cloud $\hat{\mathcal{P}}$ with the canonical point cloud $\mathcal{P}^{\mathrm{cam}}$. As suggested by our ablation studies in Section 4.4, this pointmap is an entangled representation: while it can solve for a complete pose, its rotation estimate is less accurate than the one from the pure ray branch. We therefore enforce our decoupling strategy by using only the translation component $\hat{\mathbf{t}}_{\mathrm{point}}$ as the whole model's prediction $\hat{\mathbf{t}}$ from this solved transformation.

3.3 Data Augmentation with Novel View Synthesis

Similar to prior APR methods [12, 31, 32, 40], we leverage NVS to augment our limited training datasets by rendering novel views to expand scene coverage. We specifically use 3D Gaussian Splatting (3DGS) [29] for this task due to its fast and efficient rendering. Our process is performed offline: we first train a 3DGS model on the real training images, and then use it to render a large, fixed set of synthetic views. Our final training data thus consists of the original real images and the much larger synthetic set. In each training epoch, we iterate through the synthetic set once while repeatedly cycling through the smaller real set.

Training on synthetic data introduces a domain gap between the rendered and real images. To bridge this gap, we employ domain-adversarial adaptation [20,

55]. As shown in Figure 3, a Gradient Reversal Layer (GRL) is applied to the global feature $\mathbf{f}_{global}$ from each branch. This feature is then fed to a domain classifier that is trained to discriminate between the real and synthetic domains. The GRL forces the backbones to learn domain-invariant features that can "fool" this classifier. This allows our model to benefit from training on the massive set of augmented views while maintaining its ability to generalize to real-world images.

3.4 Training and Objective Functions

Our model is trained end-to-end with a composite loss function that provides comprehensive supervision on both the final pose estimates and the intermediate geometric representations. Following a warmup phase, we switch the norm p of loss from 1 to 2 to refine the predictions more precisely.

Pose Supervision Loss. We supervise both recovered poses against the ground truth pose $\mathbf{T}_{gt} = [\mathbf{R}_{gt}|\mathbf{t}_{gt}]$. The loss combines rotation and translation errors:

$$\mathcal{L}_{pose} = w^{\mathrm{r}}_{pose} d_g(\hat{\mathbf{R}}, \mathbf{R}_{gt})^p + w^{\mathrm{p}}_{pose} \|\hat{\mathbf{t}} - \mathbf{t}_{gt}\|_p \tag{2}$$

where d_g is the geodesic distance between two rotations.

Geometry Loss. To ensure our intermediate representations are geometrically meaningful, we directly supervise them against their ground-truth counterparts. The loss is a sum of two terms: (1) a ray term that encourages alignment between the predicted rays $\hat{\mathcal{D}}$ and ground-truth rays $\mathcal{D}_{gt}$ using cosine similarity, and (2) a point term that penalizes the distance between the predicted points $\hat{\mathcal{P}}$ and the ground-truth points $\mathcal{P}_{gt}$:

$$\mathcal{L}_{geo} = w^{\mathrm{r}}_{geo} \left(1 - \cos(\hat{\mathcal{D}}, \mathcal{D}_{gt})\right) + w^{\mathrm{p}}_{geo} \|\hat{\mathcal{P}} - \mathcal{P}_{gt}\|_p \tag{3}$$

Geometric Regularization Loss. We introduce two regularization terms to enforce geometric smoothness across the predicted fields, in order to: (1) preserve local angular relationships and (2) preserves pairwise distances for local rigidity:

$$\begin{aligned} \mathcal{L}_{\mathrm{reg}} = \frac{1}{|\mathcal{N}|} \sum_{(i,j)\in\mathcal{N}} & w^{\mathrm{r}}_{reg} \left| (\hat{\mathbf{d}}_i \cdot \hat{\mathbf{d}}_j) - (\mathbf{d}^{\mathrm{cam}}_i \cdot \mathbf{d}^{\mathrm{cam}}_j) \right|_p \\ & + w^{\mathrm{p}}_{reg} \left| \|\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j\| - \|\mathbf{p}^{\mathrm{gt}}_i - \mathbf{p}^{\mathrm{gt}}_j\| \right|_p \end{aligned} \tag{4}$$

where $\mathcal{N}$ denotes the set of spatial patch neighbors.

Domain Adaptation Loss. We follow the domain-adversarial training via gradient reversal [20] on both ray and point cloud branches. The loss for a given batch is the sum of the Binary Cross-Entropy (BCE) losses:

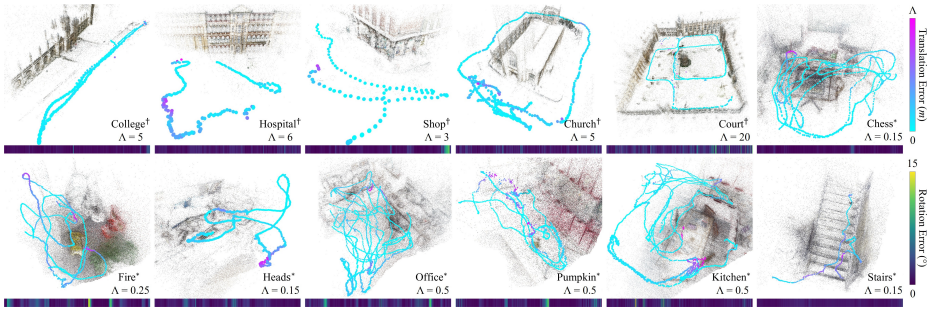

Fig. 4: Visualization of GRLoc's predictions. †: Cambridge Landmarks [28]; ∗: 7-Scenes [50]. Λ indicates scene-specific upper-limit of translation error. Sequential error bars show rotation errors.

$$\mathcal{L}_{\text{domain}} = \sum_{b \in \{\text{ray,point}\}} \mathcal{L}_{\text{BCE}}(D^b(\text{GRL}(\mathbf{f}^b_{global})), y) \tag{5}$$

where D^b is the domain classifier for a branch, and y is the domain label (e.g., 0 for synthetic, 1 for real).

Total Loss. The final training objective is a weighted sum of these components, calculated for both a batch of synthetic data ($\mathcal{L}_{syn}$) and a batch of real data ($\mathcal{L}_{real}$), along with the domain adaptation loss:

$$\begin{aligned} \mathcal{L}_{total} =& w_{syn}\mathcal{L}^{syn} + w_{real}\mathcal{L}^{real} \\ &+ w_{domain}(\mathcal{L}^{syn}_{domain} + \mathcal{L}^{real}_{domain}) \end{aligned} \tag{6}$$

where $\mathcal{L}^{syn/real} = \mathcal{L}_{pose} + \mathcal{L}_{geo} + \mathcal{L}_{reg}$.

4 Experiments

We discuss our experimental setup, implementation details, and results in this section. As discussed, our GRR paradigm aligns with APR in its end-to-end design and primary training objective. The key difference is our intermediate regression of geometric representations. Therefore, we categorize our method with APR for our main comparisons.

4.1 Experimental Setup

Datasets. In our experiments, we use two public datasets for evaluation: (1) the Cambridge Landmarks dataset [28], which consists of large-scale outdoor urban scenes ranging approximately $875 \sim 9200m^2$, and (2) the 7-Scenes dataset [50], which consists of small-scale indoor environments with spatial extents ranging $1 \sim 18m^2$. For fair comparisons with APR methods, we follow the setup of recent works [12, 31, 32] to sample subsets of the training data.

Fig. 5: Visual comparisons on 7-Scenes dataset [50]: top-right corners show ground truth, and bottom-left corners show estimated views.

Baselines. Our primary comparison is with state-of-the-art Absolute Pose Regression (APR) methods, including PMNet [32], DFNet [12], RAP [31], and Marepo [11]. For a comprehensive evaluation, we also report results against leading RPR, SCR, and PPR methods.

While our main focus is single-frame localization, we also test an optionally refined model $\text{GRLoc}_{\text{ref}}$ to ensure a fair comparison against high-accuracy SCR and PPR methods, which include refinement or non-differentiable solver steps. We adopt a recent pose-refinement strategy [31, 34] and follow the exact refinement setup of RAP [31].

Implementation Details. All of our experiments are conducted on a machine with 32 AMD EPYC 9654 CPU cores and a single NVIDIA H200 GPU.

Our feature extractor's backbone is the Swin Transformer V2 large [37], pretrained on ImageNet [14], to leverage its multi-level feature maps. We empirically set the patch grid shape to 16×16, which provides the best balance across different grid resolutions for the discussed density-performance trade-off.

For loss weights, our preliminary tests suggest using fixed weights can cause specific terms to dominate depending on diverse scene scales. We thus employ a scale-adaptive weighting strategy to ensure balanced gradient flow, initializing weights to normalize the initial magnitude to 1.0.

For 3DGS-based data augmentation, we use the gsplat library [58] and follow the 3DGS-MCMC [30] method. We generate the synthetic dataset with sampled novel camera poses by perturbing the original training views.

4.2 Evaluations on Benchmarks

Figures 4 and 5 show qualitative results on benchmarks. Additional visualizations are provided in supplement.

On the room-scale indoor scenes of the 7-Scenes dataset (using SfM pseudo ground truth [6]), our approach achieves state-of-the-art performance among APR/GRR methods, as shown in Table 1. Notably, our GRLoc outperforms Marepo [11] on average and in 5 scenes, though Marepo is trained with additional priors from the Map-Free dataset [2].

On the large-scale outdoor scenes of the Cambridge Landmarks dataset, GRLoc significantly outperforms previous APR methods, with an average reduction of over 33% in translation error and 46% in rotation error, as shown in

Table 1: Median translation (cm) and rotation (°) errors on the 7-Scenes dataset [50]. Bold: best results; Underline: second-best.

Category	Methods	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Average
APR	PoseNet [28]	10/4.02	27/10.00	18/13.00	17/5.97	19/4.67	22/5.91	35/10.50	21/7.74
	MapNet [9]	13/4.97	33/9.97	19/16.7	25/9.08	28/7.83	32/9.62	43/11.8	28/10.00
	PAE [49]	13/6.61	24/12.00	14/13.00	19/8.58	17/7.28	18/8.89	30/10.30	19/9.52
	MS-Transformer [48]	11/6.38	23/11.5	13/13.0	18/8.14	17/8.42	16/8.92	29/10.30	18/9.51
	DFNet [12]	3/1.12	6/2.30	4/2.29	6/1.54	7/1.92	7/1.74	12/2.63	6/1.93
	RAP [31]	2/0.85	6/2.84	4/4.52	4/1.57	3/1.10	5/1.10	10/1.30	5/1.90
	Marepo [11]	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
(GRR)	**GRLoc (ours)**	**0.97/0.31**	2.93/0.95	**1.40/0.90**	5.08/1.05	**2.28/0.53**	**3.13/0.69**	**3.42/0.82**	**2.75/0.75**
RPR	ExReNet [57]	5/1.63	7/2.54	3/2.71	6/1.75	7/2.04	7/2.10	19/4.87	8/2.52
	Reloc3r [16]	3/0.88	3/0.81	1/0.95	4/0.88	6/1.10	4/1.26	7/1.26	4/1.02
SCR	DSAC [8]	0.5/0.17	0.8/0.28	0.5/0.33	1.2/0.34	1.2/0.28	0.7/0.21	2.7/0.78	1.1/0.34
	ACE [5]	0.5/0.18	0.8/0.33	0.6/0.34	1.0/0.29	1.0/0.22	0.8/0.20	2.9/0.81	1.1/0.34
	GLACE [56]	0.6/0.18	0.9/0.34	0.5/0.33	1.1/0.29	0.9/0.23	0.8/0.20	3.2/0.93	1.2/0.36
PPR	FQN-MN [22]	4.1/1.31	10.5/2.97	7.63/5.86	3.6/2.36	4.6/1.76	16.1/4.42	139.5/34.67	28/7.3
	CrossFire [39]	1/0.4	3/2.3	5/1.9	5/1.6	3/0.8	2/0.8	12/1.9	4.4/1.38
	MCLoc [54]	2/0.8	3/1.4	2/1.45	4/1.3	5/1.6	6/1.6	6/2.0	4.1/1.43
	HR-APR [35]	2/0.55	2/0.75	3/1.3	2/0.64	2/0.62	2/0.67	5/1.30	2.4/0.85
	DFNet + NeFeS_{50} [10]	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
	NeRFMatch [63]	0.95/0.30	1.11/0.41	1.34/0.92	3.09/0.87	2.21/0.60	1.03/0.28	9.26/1.74	2.71/0.73
	ACE + GS-CPR [34]	0.5/0.15	0.4/0.28	0.6/0.25	0.9/0.26	1.0/0.23	0.7/0.17	1.4/0.42	0.8/0.25
	STDLoc [24]	0.46/0.15	0.57/0.24	0.45/0.26	0.86/0.24	0.93/0.21	0.63/0.19	1.42/0.41	0.76/0.24
	RAP_{ref} [31]	0.33/0.11	0.51/0.21	0.39/0.27	**0.57**/0.16	0.81/0.20	0.45/0.12	1.11/0.32	0.60/0.20
	$\text{GRLoc}_{\text{ref}}$ (ours)	**0.28/0.10**	**0.32/0.13**	**0.22/0.14**	0.65/**0.15**	**0.70/0.15**	**0.35/0.10**	**0.61/0.19**	**0.45/0.14**

Table 2. Our method also demonstrates strong performance on the challenging *Court* scene, a large-scale environment that is often omitted by prior APR work.

The results also demonstrate that our base model is already competitive with or superior to several PPR methods on both datasets. When applying the optional refinement step, $\text{GRLoc}_{\text{ref}}$ achieves results comparable to RAP_{ref} [31]. This is expected, as we adopt the same refinement module. We observe that while our unrefined GRLoc outperforms the unrefined RAP, both methods provide a sufficiently accurate initial pose for the refinement module to converge to a similar, high-quality result. During these experiments, we also noted that the refinement performance is highly dependent on the quality of the depth-enabled 3DGS model: the fidelity of its rendered RGB influences the 2D feature matching, while the accuracy of its rendered depth directly impacts the final 2D-to-3D correspondence estimation.

Interestingly, while $\text{GRLoc}_{\text{ref}}$ outperforms STDLoc [24] on 7-Scenes, it performs slightly worse on Cambridge. We attribute this to the challenges of large-scale outdoor environments, which are not only more difficult for the pose estimator itself, but also for training the high-fidelity NVS models. As discussed, the refinement quality is highly dependent on the NVS model's fidelity, and it is more difficult to achieve this in large outdoor scenes.

4.3 Inference Latency and Real-Time Feasibility

To evaluate the computational efficiency of our approach, we measure the inference latency in an online setting with a batch size of 1. Our GRLoc achieves an average inference time of 36.78 ms per frame. While slower than the highly lightweight RAP [31] (12.84 ms), it remains significantly faster than Marepo [11] (68.50 ms). The increased latency compared to RAP is primarily attributed to the high-capacity Swin Transformer backbone [37], which is computationally heavier but critical for driving our accuracy gains. Despite this heavy backbone, GRLoc maintains a competitive speed that satisfies the requirements of most real-time applications, such as augmented reality and autonomous navigation.

Table 2: Median translation (cm) and rotation (°) errors on the Cambridge [28]. Bold: best results; Underline: second-best.

	Methods	College	Hospital	Shop	Church	Average	Court
	PoseNet [28]	166/4.86	262/4.90	141/7.18	245/7.95	204/6.23	683/3.50
	MapNet [9]	107/1.89	194/3.91	149/4.22	200/4.53	163/3.64	N/A
	PAE [49]	90/1.49	207/2.58	99/3.88	164/4.16	140/3.03	N/A
APR	MS-Trans. [48]	83/1.47	181/2.39	86/3.07	162/3.99	128/2.73	N/A
	DFNet [12]	73/2.37	200/2.98	67/2.21	137/4.03	119/2.90	217/4.11
	PMNet [32]	68/1.97	103/1.31	58/2.10	133/3.73	90/2.27	N/A
	RAP [31]	52/0.90	87/1.21	33/1.48	53/1.52	56/1.28	115/1.68
(GRR)	**GRLoc (ours)**	**34/0.42**	**54/0.88**	**15/0.48**	**46/0.93**	**37/0.68**	**88/0.53**
RPR	ExReNet [57]	233/2.48	354/3.49	72/2.41	230/3.72	222/3.03	979/4.46
	Reloc3r [16]	42/0.36	62/0.55	13/0.58	34/0.58	38/0.52	122/0.73
	DSAC [8]	18/0.3	21/0.4	5/0.3	15/0.6	15/0.4	34/0.2
SCR	ACE [5]	28/0.4	31/0.6	5/0.3	18/0.6	21/0.5	43/0.2
	GLACE [56]	19/0.3	17/0.4	4/0.2	9/0.3	12/0.3	19/0.1
	LENS [40]	34/0.54	45/0.96	28/1.66	54/1.66	40/1.21	N/A
	FQN-MN [22]	28/0.38	54/0.82	13/0.63	58/2.00	38/0.96	4253/9.16
	CrossFire [39]	47/0.7	43/0.7	20/1.2	39/1.4	37/1.0	N/A
	NeFeS_{50} [10]	37/0.54	52/0.88	15/0.53	37/1.14	35/0.77	N/A
	HR-APR [35]	36/0.58	53/0.89	13/0.51	38/1.16	35/0.78	N/A
PPR	MCLoc [54]	31/0.42	39/0.73	12/0.45	26/0.88	27/0.62	N/A
	NeRFMatch [63]	12.5/0.23	20.9/0.38	8.4/0.40	10.9/0.35	14.5/0.29	19.6/0.09
	$\text{ACE}_{\text{GS-CPR}}$ [34]	20/0.29	21/0.40	5/0.24	13/0.40	15/0.33	N/A
	$\text{DFNet}_{\text{ref}}$ [31]	16/0.24	21/0.41	8/0.42	10/0.26	14/0.33	25/0.13
	STDLoc [24]	**15.0/0.17**	**11.9/0.21**	**3.0/0.13**	**4.7/0.14**	**10.1/0.14**	**15.7/0.06**
	RAP_{ref} [31]	**15**/0.23	18/0.38	5/0.23	9/0.23	12/0.27	22/0.15
	$\text{GRLoc}_{\text{ref}}$ (ours)	16/0.21	16/0.35	5/0.20	8/0.23	11/0.25	21/0.11

4.4 Ablation Study

We conduct our ablation studies on the *Shop* scene from the Cambridge Landmarks dataset to evaluate different model designs. The results are shown in Table 3. We define the following experimental conditions:

- **ViT** replaces the Swin Transformer backbone with a vanilla ViT L-16 [17].
- **W/o DA** removes the domain adaptation components during training.
- **APR** uses the same architecture as the full model, but is trained to directly regress the 6-DoF pose from the global features following standard APR.
- **GR-only** trains the network using only the geometric representation (raymap and pointmap) supervision, without utilizing the final 6-DoF pose as a direct loss signal. The final pose is purely recovered by the solver during inference.
- **W/o point** and **W/o ray** keep only the ray or point branch, respectively. **W/o point** adds a MLP head to regress the 3D translation, as the ray branch cannot solve for it.
- **Share** uses a shared feature extractor for both heads.

As shown in Table 3, we first validate our architectural choices. A Swin Transformer backbone with multi-level feature fusion successfully outperforms the **ViT** backbone. Furthermore, the significant performance drop in the **W/o DA** highlights the necessity of domain adaptation for robust feature extraction.

To validate our core formulation, we compare different supervision paradigms: **APR** (pure pose supervision), **GR-only** (pure geometric supervision), and our full model. The results indicate that pairing explicitly learned geometric representations with final pose supervision yields the optimal performance.

Table 3: Ablation studies on the *Shop* scene. Median translation (cm) and rotation (°) errors are reported.

ViT	29/0.98	**W/o DA**	33/1.25	**APR**	39/1.67	**GR-only**	20/3.96
W/o point	41/0.66	**W/o ray**	**15**/0.86	**Share**	17/0.77	**GRLoc (ours)**	**15/0.48**

Beyond the supervision paradigm, finding a robust representation for both rotation and translation is critical. The comparison between **W/o point**, **W/o ray**, **Share**, and our full **GRLoc** model traces the development of our approach:

1. We first considered a ray-only model (**W/o point**), which provides a pure representation for rotation. However, this model lacks a geometric basis for translation, forcing us to add a direct regression head that performs poorly.
2. We then tested a point-only model (**W/o ray**), with pointmap as an entangled representation. This model achieved better translation, but its rotation error was worse than the ray-only variant. This confirms our hypothesis that an entangled representation is suboptimal for recovering rotation.
3. Next, we combined both representations in a **Share** model using a single backbone, outperformed the single-branch variants but was still suboptimal. We found that the two competing objectives were "pulling against" each other, forcing the shared features to be a poor compromise for both tasks.

This led to our final GRLoc model that fully decouples the two branches. This design allows each branch to extract features specialized for its unique objective, leading to the best performance. These observations validate our claim: it is critical to decouple rotation and translation, occurring at both the representation level (by using a pure geometric representation like ray directions for rotation) and the architectural level (by using separate network branches to prevent competing objectives from degrading the features).

We believe the conflict stems from the contradictory nature of required visual features: rotation relies on translation-invariant global cues (e.g., vanishing points), while translation requires parallax-sensitive local cues (e.g., relative scale). In entangled backbones, the network must learn features that are simultaneously invariant to and sensitive to position, causing the model to "cheat" by using translation shifts to compensate for rotational drift.

5 Conclusion

In this paper, we present a reformulation of absolute pose estimation as Geometric Representation Regression (GRR), a new paradigm that regresses explicit geometric components and then analytically computes the 6-DoF pose using a differentiable solver. We demonstrate that decoupling the rotation and translation predictions is critical for resolving their competing optimization objectives and achieving high accuracy. Our GRLoc achieves state-of-the-art performance, validating that this hybrid, geometry-centric approach provides a more robust, interpretable, and generalizable path forward for visual localization.

References

1. Abouelnaga, Y., Bui, M., Ilic, S.: Distillpose: Lightweight camera localization using auxiliary learning. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 7919–7924. IEEE (2021)
2. Arnold, E., Wynn, J., Vicente, S., Garcia-Hernando, G., Monszpart, A., Prisacariu, V., Turmukhambetov, D., Brachmann, E.: Map-free visual relocalization: Metric pose relative to a single image. In: European Conference on Computer Vision. pp. 690–708. Springer (2022)
3. Balntas, V., Li, S., Prisacariu, V.: Relocnet: Continuous metric learning relocalisation using neural nets. In: Proceedings of the European conference on computer vision (ECCV). pp. 751–767 (2018)
4. Blanton, H., Workman, S., Jacobs, N.: A structure-aware method for direct pose estimation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2019–2028 (2022)
5. Brachmann, E., Cavallari, T., Prisacariu, V.A.: 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)
6. Brachmann, E., Humenberger, M., Rother, C., Sattler, T.: 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)
7. Brachmann, E., Krull, A., Nowozin, S., Shotton, J., Michel, F., Gumhold, S., Rother, C.: Dsac-differentiable ransac for camera localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6684–6692 (2017)
8. Brachmann, E., Rother, C.: 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)
9. Brahmbhatt, S., Gu, J., Kim, K., Hays, J., Kautz, J.: Geometry-aware learning of maps for camera localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2616–2625 (2018)
10. Chen, S., Bhalgat, Y., Li, X., Bian, J.W., Li, K., Wang, Z., Prisacariu, V.A.: 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 (2024)
11. Chen, S., Cavallari, T., Prisacariu, V.A., Brachmann, E.: Map-relative pose regression for visual re-localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20665–20674 (2024)
12. Chen, S., Li, X., Wang, Z., Prisacariu, V.A.: Dfnet: Enhance absolute pose regression with direct feature matching. In: European Conference on Computer Vision. pp. 1–17. Springer (2022)
13. Chen, S., Wang, Z., Prisacariu, V.: Direct-posenet: Absolute pose regression with photometric consistency. In: 2021 International Conference on 3D Vision (3DV). pp. 1175–1185. IEEE (2021)
14. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
15. Ding, M., Wang, Z., Sun, J., Shi, J., Luo, P.: Camnet: Coarse-to-fine retrieval for camera re-localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2871–2880 (2019)

16. Dong, S., Wang, S., Liu, S., Cai, L., Fan, Q., Kannala, J., Yang, Y.: Reloc3r: Large-scale training of relative camera pose regression for generalizable, fast, and accurate visual localization. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 16739–16752 (2025)
17. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
18. Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T.: 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)
19. Fischler, M.A., Bolles, R.C.: 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)
20. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., Lempitsky, V.: Domain-adversarial training of neural networks. Journal of machine learning research **17**(59), 1–35 (2016)
21. Gao, X.S., Hou, X.R., Tang, J., Cheng, H.F.: Complete solution classification for the perspective-three-point problem. IEEE transactions on pattern analysis and machine intelligence **25**(8), 930–943 (2003)
22. Germain, H., DeTone, D., Pascoe, G., Schmidt, T., Novotny, D., Newcombe, R., Sweeney, C., Szeliski, R., Balntas, V.: 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)
23. Giang, K.T., Song, S., Jo, S.: Learning to produce semi-dense correspondences for visual localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19468–19478 (2024)
24. Huang, Z., Yu, H., Shentu, Y., Yuan, J., Zhang, G.: From sparse to dense: Camera relocalization with scene-specific detector from feature gaussian splatting. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 27059–27069 (2025)
25. Humenberger, M., Cabon, Y., Guerin, N., Morat, J., Leroy, V., Revaud, J., Rerole, P., Pion, N., De Souza, C., Csurka, G.: Robust image retrieval-based visual localization using kapture. arXiv preprint arXiv:2007.13867 (2020)
26. Jiang, X., Wang, F., Galliani, S., Vogel, C., Pollefeys, M.: R-score: Revisiting scene coordinate regression for robust large-scale visual localization. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 11536–11546 (2025)
27. Kabsch, W.: A solution for the best rotation to relate two sets of vectors. Foundations of Crystallography **32**(5), 922–923 (1976)
28. Kendall, A., Grimes, M., Cipolla, R.: 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)
29. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
30. Kheradmand, S., Rebain, D., Sharma, G., Sun, W., Tseng, Y.C., Isack, H., Kar, A., Tagliasacchi, A., Yi, K.M.: 3d gaussian splatting as markov chain monte carlo. In: Advances in Neural Information Processing Systems (NeurIPS) (2024), spotlight Presentation

31. Li, S., Tan, S., Chang, B., Zhang, J., Feng, C., Li, Y.: Adversarial exploitation of data diversity improves visual localization. In: International Conference on Computer Vision (ICCV) (2025)
32. Lin, J., Gu, J., Wu, B., Fan, L., Chen, R., Liu, L., Ye, J.: Learning neural volumetric pose features for camera localization. In: European Conference on Computer Vision. pp. 198–214. Springer (2024)
33. Lindenberger, P., Sarlin, P.E., Pollefeys, M.: Lightglue: Local feature matching at light speed. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 17627–17638 (2023)
34. Liu, C., Chen, S., Bhalgat, Y.S., HU, S., Cheng, M., Wang, Z., Prisacariu, V.A., Braud, T.: GS-CPR: Efficient camera pose refinement via 3d gaussian splatting. In: The Thirteenth International Conference on Learning Representations (2025)
35. Liu, C., Chen, S., Zhao, Y., Huang, H., Prisacariu, V., Braud, T.: 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 (2024)
36. Liu, L., Li, H., Dai, Y.: Efficient global 2d-3d matching for camera localization in a large-scale 3d map. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2372–2381 (2017)
37. Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al.: Swin transformer v2: Scaling up capacity and resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12009–12019 (2022)
38. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
39. Moreau, A., Piasco, N., Bennehar, M., Tsishkou, D., Stanciulescu, B., de La Fortelle, A.: 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)
40. Moreau, A., Piasco, N., Tsishkou, D., Stanciulescu, B., de La Fortelle, A.: Lens: Localization enhanced by nerf synthesis. In: Conference on Robot Learning. pp. 1347–1356. PMLR (2022)
41. Pietrantoni, M., Csurka, G., Sattler, T.: Gaussian splatting feature fields for (privacy-preserving) visual localization. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 1082–1092 (2025)
42. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: 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)
43. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: 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)
44. Sarlin, P.E., Unagar, A., Larsson, M., Germain, H., Toft, C., Larsson, V., Pollefeys, M., Lepetit, V., Hammarstrand, L., Kahl, F., et al.: 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. pp. 3247–3257 (2021)
45. Sattler, T., Leibe, B., Kobbelt, L.: Fast image-based localization using direct 2d-to-3d matching. In: 2011 International Conference on Computer Vision. pp. 667–674. IEEE (2011)

46. Sattler, T., Leibe, B., Kobbelt, L.: Efficient & effective prioritized matching for large-scale image-based localization. IEEE transactions on pattern analysis and machine intelligence **39**(9), 1744–1756 (2016)
47. Sattler, T., Zhou, Q., Pollefeys, M., Leal-Taixe, L.: 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)
48. Shavit, Y., Ferens, R., Keller, Y.: Learning multi-scene absolute pose regression with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2733–2742 (2021)
49. Shavit, Y., Keller, Y.: Camera pose auto-encoders for improving pose regression. In: European Conference on Computer Vision. pp. 140–157. Springer (2022)
50. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: 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)
51. Sidorov, G., Mohrat, M., Gridusov, D., Rakhimov, R., Kolyubin, S.: Gsplatloc: Grounding keypoint descriptors into 3d gaussian splatting for improved visual localization. In: 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 12601–12607. IEEE (2025)
52. 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. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7199–7209 (2018)
53. Tang, S., Tang, C., Huang, R., Zhu, S., Tan, P.: Learning camera localization via dense scene matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1831–1841 (2021)
54. Trivigno, G., Masone, C., Caputo, B., Sattler, T.: 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)
55. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7167–7176 (2017)
56. Wang, F., Jiang, X., Galliani, S., Vogel, C., Pollefeys, M.: Glace: Global local accelerated coordinate encoding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21562–21571 (2024)
57. Winkelbauer, D., Denninger, M., Triebel, R.: Learning to localize in new environments from synthetic training data. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 5840–5846. IEEE (2021)
58. Ye, V., Li, R., Kerr, J., Turkulainen, M., Yi, B., Pan, Z., Seiskari, O., Ye, J., Hu, J., Tancik, M., et al.: gsplat: An open-source library for gaussian splatting. Journal of Machine Learning Research **26**(34), 1–17 (2025)
59. Yen-Chen, L., Florence, P., Barron, J.T., Rodriguez, A., Isola, P., Lin, T.Y.: 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)
60. Zeisl, B., Sattler, T., Pollefeys, M.: Camera pose voting for large-scale image-based localization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2704–2712 (2015)
61. Zhang, Y., Zhao, X.: Semantic-guided camera ray regression for visual localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 25639–25648 (2025)

62. Zhao, B., Yang, L., Mao, M., Bao, H., Cui, Z.: Pnerfloc: Visual localization with point-based neural radiance fields. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 7450–7459 (2024)
63. Zhou, Q., Maximov, M., Litany, O., Leal-Taixé, L.: The nerfect match: Exploring nerf features for visual localization. In: European Conference on Computer Vision. pp. 108–127. Springer (2024)