Semantic-guided Camera Ray Regression for Visual Localization

Yesheng Zhang, Xu Zhao*
School of Automation and Intelligent Sensing, Shanghai Jiao Tong University
{preacher, zhaoxu}@sjtu.edu.cn

Abstract

This work presents a novel framework for Visual Localization (VL), that is, regressing ***camera rays*** *from query images to derive camera poses. As an overparameterized representation of the camera pose, camera rays possess superior robustness in optimization. Of particular importance, Camera Ray Regression (CRR) is privacy-preserving, rendering it a viable VL approach for real-world applications. Thus, we introduce DINO-based Multi-Mappers, coined DIMM, to achieve VL by CRR. DIMM utilizes DINO as a scene-agnostic encoder to obtain powerful features from images. To mitigate ambiguity, the features integrate both local and global perception, as well as potential geometric constraint. Then, a scene-specific mapper head regresses camera rays from these features. It incorporates a semantic attention module for soft fusion of multiple mappers, utilizing the rich semantic information in DINO features. In extensive experiments on both indoor and outdoor datasets, our methods showcase impressive performance, revealing a promising direction for advancements in VL.*

1. Introduction

Visual Localization (VL) involves estimating 6-degree-of-freedom camera pose of an image taken from a known scene, which is also known as camera relocalization. VL plays a vital role in many vision, robotic and graphic applications, such as augment reality, virtual reality and autonomous driving. Despite extensive research, achieving precise, robust, and privacy-preserving VL [28, 38] continues to present a significant challenge.

Current VL methods can be categorized into *explicit* and *implicit* methods based on their mapping approaches. Traditional *explicit* methods [2, 30, 31] involve an *explicit* 3D map, *e.g*., point clouds. They establish 2D-3D (image-to-map) correspondences by Feature Matching, and then estimate camera pose via PnP and RANSAC [10, 12]. While advanced methods [11, 39] leads to high accuracy, the storage demands of explicit maps are troublesome especially in large-scale scenes. Additionally, explicit maps raise privacy concerns, restricting their applications in real-world settings.

* Corresponding author.

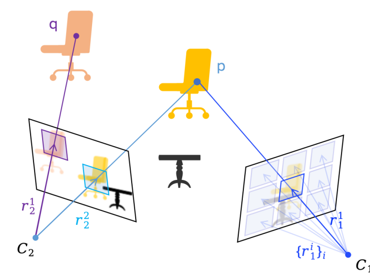

Figure 1. **The ambiguity challenge in Camera Ray Regression (CRR) for VL.** Effective global and local disambiguations are critical for CRR. Global ambiguity arises from rays of different patches (cf. r_2^1 and r_2^2 in the figure), while local ambiguity occurs within the same patch but from different viewing angles (cf. r_1^1 and r_2^2). Semantic information, *e.g*., the table in the figure to differentiate between chairs, along with geometry constraints (geometry properties of rays $\{r_1^i\}_i$) are essential for the disambiguations.

Implicit methods, on the other hand, leverage learning models as implicit and light-weight maps, which mainly include two frameworks: Absolute Pose Regression (**APR**) and Scene Coordinate Regression (**SCR**). APR directly utilizes networks to regress camera poses (rotations and translations) from images [6, 15], offering enhanced privacy protection. However, the compact nature of absolute pose makes it vulnerable to noise, thereby leading to accuracy issue [6].

To improve precision, employing an overparameterized representation for the camera pose is crucial [35]. Thus, SCR framework [22, 41] is proposed, relying on 2D-3D matches as an indirect yet robust camera pose representation, akin to its explicit counterparts. ACE [4] is a milestone within this framework, utilizing simple networks to regress pixel-aligned scene coordinates. It introduces a scene-centric patch-level training strategy, *i.e*., Gradient Decorrelation Training (**GDT**), enabling both efficient and precise mapping. Nevertheless, the unbounded 3D point error is unstable in training, necessitating manual hyperparameter tuning or additional outlier rejection modules [5, 20]. Besides, SCR methods still run the risk of 3D privacy breaches, as they

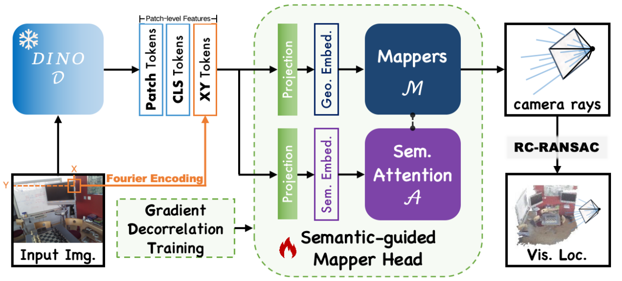

Figure 2. **The proposed DIMM approach performing CRR for VL.** This approach utilizes DINO as a scene-agnostic feature encoder to output carefully designed patch-level features. Then, a scene-specific mapper head integrates multi-mapper results under semantic guidance, through a semantic attention module. After the scene-centric training technology, DIMM is capable of predicting camera ray parameters for accurate and privacy-preserving VL.

disclose detailed 3D scene information [29].

Recently, a promising camera pose representation, camera ray [27, 35], has garnered increased attention in the vision field. Its overparameterization is beneficial for patch-level learning [23, 44], aligning well with the effective GDT in VL. This inspires us to introduce camera rays in the VL tasks. In particular, we propose adopting a learning model to regress patch-level camera rays from the image, termed as Camera Ray Regression (CRR). Then, the camera pose can be achieved by solving two linear problems [44]. This implementation of VL could strike a balance between efficacy and privacy preservation. Firstly, it belongs to the implicit mapping category, incurring low storage costs. As a overparameterized pose representation, it can yield improved precision [35] than APR. Also, the ray error is more stable than the unbounded 3D point error [27] in optimization of model training. Hence, CRR obviates the need for intricate training setting like SCR methods [5, 20, 41]. More importantly, it is hard to recover 3D scene information from camera rays, leading to a privacy-preserving VL implementation.

However, challenges exist in introducing camera rays in VL, primarily stemming from the issue of *ambiguity* (cf. Fig. 1). This ambiguity can generally be divided into two levels. The first is *global ambiguity*, referring to distinguish rays of different image patches, particularly in repetitive or textureless scenes. The similar ambiguity is encountered in SCR [4, 41], where leveraging **powerful local features**, such as **semantic** features [20], has proven effective. The second is *local ambiguity*, unique to CRR, which involves differentiating rays from the *same* image patch with different view angles. The ambiguity becomes troublesome for patches with continuous depth, where changes in view direction result in minor differences in patch content but obvious variations in the corresponding rays. To eliminate this ambiguity, we emphasize two observations on the CRR task. **1)** *The entire image often contains patches with depth discontinuities.* Their appearance changes under different viewing angles are significant enough to resolve the local ambiguity. Thus, the **image-level perception** is important for CRR disambiguation, which is also proven to be beneficial for handling global repetitiveness [41]. **2)** *The rays within the same image are governed by the same camera pose.* This **geometric constraint** allows confident rays to correct uncertain ones. Meanwhile, the smoothness of the patch-ray mapping [44] makes this geometric constraint learnable. Therefore, the integration of **image-level perception**, **geometric constraint** and **robust local features** is critical for the *global and local disambiguation* of CRR.

Based on the above analysis, this work presents DINO-based Multi-Mappers (DIMM, Fig. 2), as the first implementation of CRR in VL. Following the scene-centric design of ACE, DIMM consists of a scene-agnostic encoder and a scene-specific head, but possessing several key modules *tailed for CRR*. Specifically, we utilize DINO [8] as the feature encoder, utilizing its powerful semantic perception for disambiguation. It provides **robust** patch tokens as **local features** while supporting `CLS` tokens for **image-level perception**. We further incorporate Fourier Encoding [24] of the image position for each patch into the features (`XY` tokens), as the potential **geometric constraint** of camera rays are associated with patch locations. Furthermore, we propose a scene-specific mapper head which adopts a semantic attention module to softly fuse ray regression results from multiple Multi-Layer Perceptron (MLP)-based mappers. This head effectively utilizes the inherit semantic prior of the DINO features, to enhance ray regression performance especially in large scenes. Finally, we propose a ray-level RANSAC algorithm for accurate pose estimation, decomposing the rotation and translation calculation [44].

Our contributions are summaries as follows.

1. To the best of our knowledge, we are the first to introduce the camera ray into VL, with a tailored network design to achieve query image poses by camera ray regression.
2. We propose carefully designed scene-agnostic features to handle local and global disambiguation in CRR, along with a semantic-guided mapper head that enables a soft multi-mapper ensemble.
3. Comprehensive experiments on indoor and outdoor datasets demonstrate the effectiveness of our method, achieving results that rival or surpass the existing *state-of-the-art*. Additionally, extensive ablation studies validate the contributions of each module.

2. Related Work

Feature Matching-based VL. Traditional VL methods [30, 32, 33] rely on Feature Matching (FM) to determine the position and direction of cameras. These techniques aim to match 3D scene points with 2D image points, enabling the camera pose calculation through PnP+RANSAC-based

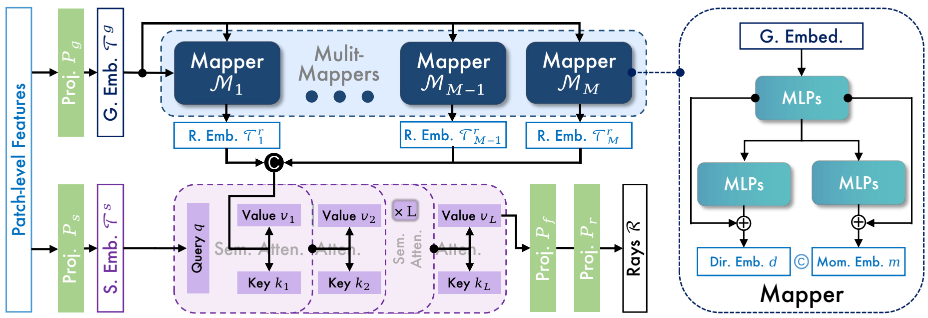

Figure 3. **The Multi-Mapper network with semantic attention as our scene-specific head.** The patch-level features are first projected to geometry and semantic embeddings. The geometry embedding is fed into multiple MLP-based mappers to derive ray embeddings. These embeddings are then fused under the guidance of the semantic embedding to achieve the camera rays, through cross attention modules.

algorithms [10, 12]. As a result, these approaches necessitate the storage of explicit maps containing 3D points and their feature descriptors. By leveraging robust matching algorithms [11, 19, 39, 42, 45], FM-based localization can deliver precise outcomes. Nevertheless, explicit maps raises concerns about privacy leaks [28]. Furthermore, as the scene size expands, managing the increasingly large explicit maps becomes challenging, leading to storage issues. In contrast, our method employs implicit maps that have lower storage requirements and privacy protection.

Scene Coordinate Regression. Similar to FM-based VL, Scene Coordinate Regression (SCR) also accomplishes camera poses through 3D-2D matching [3]. However, SCR directly employs network to regress pixel-aligned 3D scene coordinates, which addresses the map storage issue through implicit mapping. ACE [4], a prominent approach within SCR, introduces a gradient decorrelation training technique enabling mapping within 5 minutes using a 4MB network. Subsequent methods [5, 14, 20, 22, 41] built upon ACE further enhance VL accuracy mainly by solid feature selection. Nevertheless, the limited network size of ACE pose challenges in fitting large scenes, necessitating spatial segmentation of the large scene for sub-map construction. Despite its efficacy, this approach may result in ambiguous localization at the boundaries of segmented sub-scenes. At the same time, it is noteworthy that the SCR methods output the 3D information of scenes, *i.e.*, 3D coordinates, thereby leading to a potential risk of privacy leak.

Absolute Pose Regression. Absolute Pose Regression (APR) also adopts implicit mapping, but directly utilizes networks to predict the rotation and translation of cameras. However, this compact parameter representation of camera pose is highly sensitive to noises [12, 44]. Regressing pose parameters from images, thus, is challenging [15–17, 36], and the accuracy is insufficient to compare with methods using over-parameterized representations, *e.g.*, 2D-3D correspondences. Recent APR approach [6] opts to predict the camera pose based on the SCR output, resulting in favorable accuracy. However, this success is heavily reliant on the precision of SCR methods. In contrast, our method employs camera rays to depict the camera pose. This over-parameterized representation is more conducive to learning optimization [25, 27, 44], enabling superior accuracy.

Camera Ray in Vision. Recently, camera rays have emerged as a promising camera parametrization, representing a generic camera configuration [35]. In recent work [23, 43, 44], camera rays have been proven to be well-suited for patch-level learning, with applications in multiple vision tasks. Also, in 3D reconstruction works [25, 27], camera rays yield more robust geometric optimization than classical correspondence-based representations, *e.g.*, 3D-2D matches. This advantage stems from their overparameterized nature and constrained ray error. Likewise, our approach is the first to introduce camera rays in VL, performing Camera Ray Regression (CRR) to achieve camera poses from images. Its patch-level property aligns well with the Gradient Deceleration Training [4], and the bounded ray error is beneficial for optimization in VL Learning as well. Furthermore, the inherent privacy-preserving attribute of CRR positions it as a promising direction for VL implementations.

3. Method

In this section, we first present the formulation of CRR for the VL task in Sec. 3.1. Then the proposed DIMM method is described in detail. DIMM adopts a scene-centric setup [4, 14, 41] by dividing the network into a scene-agnostic encoder (Sec. 3.2) and a scene-specific head (Sec. 3.3). Additionally, the mapping process incorporates GDT [4] and utilizes a specially designed loss function (Sec. 3.4). To further enhance accuracy, we propose a RANSAC-based algorithm for estimating camera pose from rays in Sec. 3.5 .

3.1. Problem Formulation

Employing CRR for VL involves two main steps: **1)** regressing the patch-level camera ray parameters from the input image and **2)** calculating the camera pose from the ray parameters. Given an input image $I \in \mathbb{R}^{H \times W \times C}$, it is first divided uniformly into a set of patches (p_i) of size $s \times s$:

$$\{p_i \in \mathbb{R}^{s \times s \times C}\}_{i=1}^{N}, \tag{1}$$

where $N = HW/s^2$. Subsequently, the model $\mathcal{F}$ regresses the ray parameters corresponding to the center points of image patches, expressed in the world coordinate system, $\hat{\mathcal{R}} = \{r_i \in \mathbb{R}^6\}_{i=1}^{N}$.

$$\hat{\mathcal{R}} = \mathcal{F}(I), \tag{2}$$

where the ray parameters are expressed in Plücker coordinates [12]: $r_i = [d_i, m_i]$, where $d_i \in \mathbb{R}^3$ denotes the ray direction and $m_i \in \mathbb{R}^3$ represents the ray moment. Upon obtaining sufficient rays (≥ 2), the camera pose can be acquired by solving two distinct linear problems corresponding to rotation and translation, respectively [44].

Firstly, in terms of rotation, given the camera intrinsic, the Plücker coordinates of all camera rays in the camera coordinate system can be achieved, denoted as $\{r_i^c = [d_i^c, 0]\}_i^N$. The rotation matrix R, thus, corresponds to the transformation from the ray directions in the camera coordinate system to those in the world coordinate system:

$$\hat{R} = \arg \min_{||R||=1} \sum_i ||Rd_i^c - d_i||_2^2. \tag{3}$$

Subsequently, the translation, also known as the world position of the camera optical center denoted as c, is determined. Since c is the intersection point of all camera rays, the optimization problem can be formulated as:

$$\hat{c} = \arg \min_{c} \sum_i ||c \times d_i - m_i||_2^2. \tag{4}$$

Through these two linear problems, the camera pose ($[\hat{R}|\hat{c}]$) can be deduced from the camera rays.

3.2. Feature Encoder

Given the patch-wise independent GDT, it entails that the scene-specific head must learn the mapping from patch-level features to ray parameters. To ensure the learnability of this, ambiguity within patch features, particularly the local ambiguities, needs to be addressed. Thus, based on our prior analyze, patch-level features must possess both robust local representation and image-level perception. Hence, we opt for DINO [26] as our encoder backbone. From an input image I, it can generate patch-level feature tokens, $\{p_i\}_i^N$, and provide the `CLS` token for image-level perception. According to prior work [13], fine-tuning DINO's last two blocks can effectively enhance the performance of VL tasks. Consequently, we also fine-tune DINO using the ACE Encoder training method [4], resulting in $\mathcal{D}$.

$$\{\mathcal{T}_i\}_i^N, \mathcal{T}_{[\mathsf{CLS}]} = \mathcal{D}(I), \tag{5}$$

Algorithm 1 RC-RANSAC Algorithm

Input: $\hat{\mathcal{R}} = \{[\hat{d}_i, \hat{m}_i]\}_{i=1}^N$; *# camera rays from DIMM*
Output: $\hat{R} \in \mathbb{R}^{3\times3}$, $\hat{c} \in \mathbb{R}^{3\times1}$; *# camera pose*

1: Get Rotation by RANSAC: $\hat{R} \leftarrow$ *R-RANSAC*$(\{\hat{d}_i\}_{i=1}^N)$; cf. Algorithm 2 of the Suppl.
2: Ray Correction: $\hat{\mathcal{R}} \leftarrow \{[\hat{R}\hat{d}_i^c, \hat{m}_i]\}_{i=1}^N$;
3: Get Camera Center by RANSAC: $\hat{c} \leftarrow$ *C-RANSAC*$(\hat{\mathcal{R}})$; cf. Algorithm 3 of the Suppl.

Here, $\mathcal{T}_i \in \mathbb{R}^D$ represents the patch token corresponding to p_i locally, $\mathcal{T}_{[\mathsf{CLS}]} \in \mathbb{R}^D$ denotes the `CLS` token with global perception, N signifies the number of patches, and D is the token dimension.

Meanwhile, the geometric constraints of the rays from the same camera are crucial for eliminating local ambiguities. Thus, the network must also explicitly specify the spatial image location of patches during training. To further prevent the positional feature from being overshadowed by other dimensions, we draw inspiration from Nerf [24] and enhance this feature using Fourier Encoding (FE) [40].

$$\mathcal{T}_{\mathsf{xy}_i} = FE(\{x_i, y_i\}_i) \in \mathbb{R}^{N \times D_F} \tag{6}$$

Here, $\{x_i, y_i\}_i$ represents the positions of image patches $\{p_i\}_i$ in the original image, and D_F is the dimension of the Fourier encoding, set to 16. Finally, the scene-independent patch-level feature is a combination of the aforementioned three features. For each patch p_i, we have:

$$\mathcal{T}_i^* := \mathcal{T}_i \otimes \mathcal{T}_{[\mathsf{CLS}]} \otimes \mathcal{T}_{\mathsf{xy}_i} \in \mathbb{R}^{2D+D_F}, \tag{7}$$

Here, $\otimes$ denotes concatenation along the last dimension. This carefully designed feature to address ambiguity serves as the foundation of our CRR method.

3.3. Mapper Head

The scene-specific mapper head is responsible for transforming input features to ray parameters in CRR. A powerful feature encoder allows similar transformations to be achieved with simple MLPs [4, 14, 22, 41]. However, these mappers struggles to handle large scenes due to its limited capacity. To address the issue, spatial clustering is applied to divide the large scene [4, 41], with multiple MLPs fitting different sub-scenes independently. While effective, this hard scene partitioning can lead to ambiguous results near the sub-scene boundaries. In contrast, we adopt a soft map partitioning. As its core, we fuse the outputs of multiple sub-mappers with soft semantic guidance. Through end-to-end training, the sub-mappers are able to adaptively focus on different semantic regions. Particularly, we employ two MLP projection layers (P_s and P_g) to first project the input feature into a *geometric* embedding $\mathcal{T}_i^g$ and a *semantic* embedding $\mathcal{T}_i^s$, with D_h hidden dimension respectively.

Category	Method	Mapping w/ 3D info.	Mapping Time	Mapping Size
FM-based	hloc(SP+SG)	No	1.5 hours	∼2GB
SCR	DSAC*	Yes	15 hours	28MB
	ACE	No	5 minutes	4MB
	ACE (×4)	No	20 minutes	14MB
	EGFS	No	12 minutes	4.5MB
	EGFS (dual)	No	21 minutes	9MB
	D2S	Yes	∼9.4 hours	22 MB
APR	$marepo$	No	5 minutes	98.9MB
	$mareporepla_S$	No	∼15 minutes	98.9MB
CRR	DIMM (Ours)	No	∼1 hour	16.2 MB

Table 1. **The mapping size and time comparison between VL methods from different categories.** As a new flavor of VL, our DIMM achieves mapping time and memory size comparable to other methods while ensuring privacy protection.

$$\mathcal{T}^g = P_g(\mathcal{T}_i^*) \in \mathbb{R}^{D_h}, \tag{8}$$

$$\mathcal{T}^s = P_s(\mathcal{T}_i^*) \in \mathbb{R}^{D_h}, \tag{9}$$

where we omit the indices of the patches for simplification. Next, the sub-mappers $\mathcal{M} = \{\mathcal{M}_j\}_{j=1}^M$, composed of M MLPs with depth D_M, take the geometric embedding $\mathcal{T}^g$ to obtain M ray embeddings $\{\mathcal{T}_j^r\}_{j=1}^M$.

$$\mathcal{T}_j^r = \mathcal{M}_j(\mathcal{T}^g) \in \mathbb{R}^{D_h}. \tag{10}$$

Afterwards, in the sequence of L cross attention layers ($\mathcal{A} = \{\mathcal{A}_l\}_{l=1}^L$, termed as *Semantic Attention*), we replicate the semantic embedding ($\mathcal{T}_i^s$) M times along the first dimension to serve as the query q.

$$\otimes\{\mathcal{T}^s\}^M \mapsto q \in \mathbb{R}^{M\times D_h}. \tag{11}$$

All ray embeddings are concatenated along the first dimension to form both the key and value of the first layer.

$$\otimes\{\mathcal{T}_j^r\}_j^M \mapsto k_0, v_0 \in \mathbb{R}^{M\times D_h}. \tag{12}$$

By iteratively updating the key and value through the attention layers, we fuse the results of all sub-mappers by weighting them using the semantic guidance from DINO.

$$A_l(q, k_l, v_l) \mapsto v_{l+1}, k_{l+1}. \tag{13}$$

The final value is first reshaped: $v_L \in \mathbb{R}^{M\times D_h} \rightarrow \mathbb{R}^{MD_h}$, followed by projection to obtain the ultimate ray embedding: $\mathcal{T}_f^r = P_f(v_L) \in \mathbb{R}^{D_h}$. Finally, the ray parameters $\hat{r} = [\hat{d}, \hat{m}]$ are obtained through a straightforward MLP projection layer.

$$\hat{r} = P_r(\mathcal{T}_f^r) \in \mathbb{R}^6. \tag{14}$$

Based on this head, we can achieve the corresponding camera ray from a patch-level embedding from our feature encoder.

3.4. Training Loss

Adopting the patch-level Gradient Decorrelation Training, our loss function operates independently between patches. First, we utilize the L2 distance between the ground truth (r_{GT}) and predicted rays as the optimization target.

$$\mathcal{L}_{l2} = ||r_{GT} - \hat{r}||_2^2. \tag{15}$$

Besides, to enhance the model awareness of geometric constraints, we incorporate a center loss $\mathcal{L}_c$ based on the geometric properties of camera rays. The principle is any camera ray passes through a corresponding camera center [12].

$$\mathcal{L}_c = ||c \times \hat{d} - \hat{m}||_2^2. \tag{16}$$

Finally, our patch-level loss function is $\mathcal{L} = \mathcal{L}_{l2} + \alpha\mathcal{L}_c$, where α is a balance term, which is empirically set as 0.2. Our training loss is simple yet effective compared to SCR methods [4, 14, 22], benefiting from the robustness of rays in optimization.

3.5. RC-RANSAC

Similar to SCR, it is also not guaranteed that all camera rays can be accurately predicted in CRR. Therefore, employing the RANSAC mechanism to eliminate outliers is a beneficial approach. Unlike the PnP solution, when solving for camera pose from rays, the rotation and translation constitute two linear problems. Hence, the RANSAC process can be separately applied. Given that the rotation solution is independent of translation, we introduce the *RC-RANSAC* (Algorithm 1) as a modification of the general RANSAC algorithm. Specifically, after obtaining the predicted camera rays $\hat{\mathcal{R}} = \{[\hat{d}_i, \hat{m}_i]\}_i^N$, we initially use *R-RANSAC* (cf. Algorithm 2 of the Suppl.) to determine the rotation $\hat{R}$ from the ray directions only. Subsequently, all camera rays are corrected using the obtained rotation matrix $\hat{R}$ to get better rays: $\hat{\mathcal{R}}^* = \{[\hat{R}d_i^c, \hat{m}_i]\}_i$. The camera translation $\hat{c}$ is then derived from the corrected rays by another *C-RANSAC* process (cf. Algorithm 3 of the Suppl.). Ultimately, combining the rotation and translation yields the final camera pose: $[\hat{R}, \hat{c}]$ for VL.

4. Experiments

4.1. Implementation Details

Network Configuration. For the feature encoder $\mathcal{D}$, we use the `dinov2_vitb14_reg` as our backbone, with an embedding dimension of 768 ($D = 768$). The frequency of Fourier Encoding is set as 16, making $D_F = 64$. On the other hand, the mapper head consists of 4 sub-mappers and a semantic attention model with 4 attention layers. Each sub-mapper has a 4-depth MLP with skip connections. The first 2 layers form the common mapping, and the other 2 layers output direction and moment embedding respectively

Catagory	Method	Indoor-6: scene1 $(cm/°)$	scene1 $(\%)$	scene2a $(cm/°)$	scene2a $(\%)$	scene3 $(cm/°)$	scene3 $(\%)$	scene4a $(cm/°)$	scene4a $(\%)$	scene5 $(cm/°)$	scene5 $(\%.)$	scene6 $(cm/°)$	scene6 $(\%)$	Average $(cm/°)$	Average $(\%)$
FM-based	Hloc [30]	3.2/0.5	64.8	-/-	51.4	2.1/0.4	81.0	-/-	69.0	6.1/0.9	42.7	2.1/0.4	79.9	-/-	64.8
	Hloc+SLD [9]	2.9/0.4	68.7	3.4/0.6	62.7	1.9/0.3	81.0	2.8/0.5	73.9	5.4/0.8	45.3	2.1/0.4	82.0	3.1/0.5	68.9
APR	PoseNet [15]	159.0/7.5	0.0	-/-	-	141.0/9.3	0.0	-/-	-	179.3/9.4	0.0	118.2/9.3	0.0	-/-	-
SCR	DSAC* [3]	12.3/2.1	18.7	7.9/0.9	28.0	13.1/2.3	19.7	3.7/1.0	60.8	40.7/6.7	10.6	6.0/1.4	44.3	13.9/2.4	30.4
	ACE [4]	13.6/2.1	24.9	6.8/0.7	31.9	8.1/1.3	33.0	4.8/0.9	55.7	14.7/2.3	17.9	6.1/1.1	45.5	9.0/1.4	34.8
	NBE+SLD(E) [9]	7.5/1.2	28.4	7.3/0.7	30.4	6.2/1.3	43.5	4.6/1.0	54.4	6.3/1.0	37.5	5.8/1.3	44.6	6.3/1.1	39.8
	NBE+SLD [9]	6.5/0.9	38.4	7.2/0.7	32.7	4.4/0.9	53.0	3.8/0.9	66.5	6.0/**0.9**	40.0	5.0/1.0	50.5	5.5/0.9	46.9
	EGFS [20]	-/-	46.4	-/-	60.6	-/-	56.4	-/-	78.7	-/-	22.8	-/-	71.6	-/-	56.1
	EGFS-q [20]	-/-	**58.5**	-/-	59.1	-/-	**67.0**	-/-	76.7	-/-	30.6	-/-	75.9	-/-	61.3
	D2S [5]	**4.8/0.8**	51.8	4.0/**0.4**	61.1	**3.6/0.7**	60.0	**2.1/0.5**	**84.8**	**5.8/0.9**	**45.5**	**2.4/0.5**	75.2	**3.4/0.6**	63.1
CRR	DIMM	5.5/1.5	44.4	4.2/0.8	66.7	4.4/1.1	61.6	2.8/1.0	79.7	7.6/1.5	23.6	3.3/0.8	78.3	4.5/1.1	60.7
	DIMM-R	5.5/1.5	45.2	**3.3**/0.6	**77.4**	4.2/1.1	62.5	2.7/1.0	81.0	6.3/1.5	33.0	3.2/0.8	**79.8**	4.2/1.1	**63.2**

Table 2. Localization results on Indoor-6 [9]. We report the median errors in cm for the position, degree ($^\circ$) for the orientation, and recall at $5cm/5^\circ$ of the Indoor-6 dataset. The **best** results, the second best and our results are highlighted, except for the FM-based method.

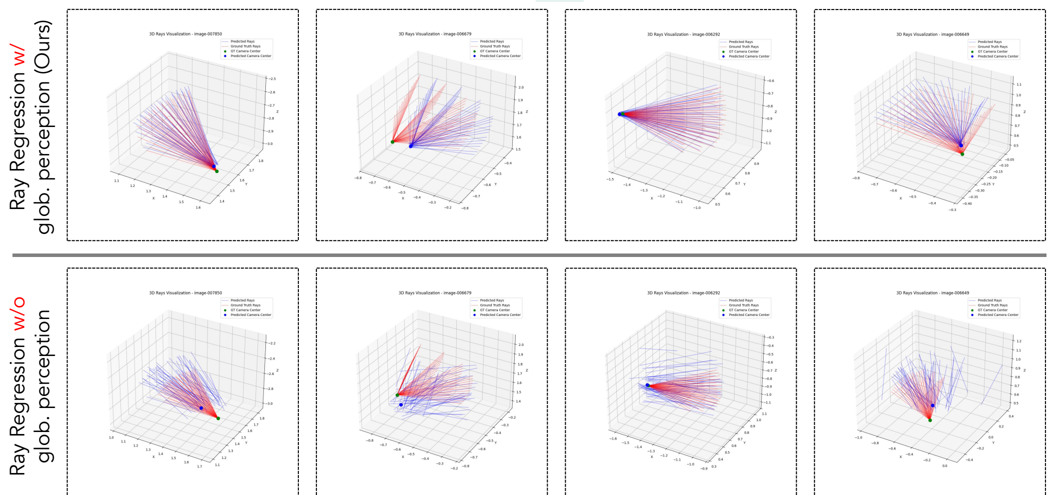

Figure 4. **The qualitative comparison between ray regression w/ or w/o global perception in the scene1 of Indoor-6.** These results contain predicted and ground truth rays from the same cameras, along with their camera centers. The samples indicate that ray predictions w/o global perception (without the `CLS` and XY tokens, bottom) lack geometric consistency, although some individual rays are accurate. In contrast, incorporating global perception facilitates learning the geometric constraint of rays, resulting in precise ray regression (top).

(cf. Fig. 3). The hidden dimension D_h is set as 256. More ablation experiments can be found in Sec. 4.3.

Training and Hardware Details. Firstly, according to related work [13], we fine-tune the last two blocks of DINO in terms of CRR task for better performance. Particularly, we follow the encoder training protocol of [4] and [13], using the first 100 training scenes of ScanNet [7] to update the weights of unfrozen DINO blocks. The training is performed on 4 NVIDIA A800 GPUs using gradient accumulation [4] in parallel. Next, the mapper head is trained on 1 NVIDIA A800 GPU for each scene, with a training buffer of 26M encoder features. The large buffer size is important for CRR training, as most of patches have different rays. We observed a *buffer size scaling law* in experiments (cf. Suppl. B). We do 20 passes over the training buffer, utilizing a batch size of 5120. The optimization uses AdamW [21] with a learning rate between $5e^{-5}$ and $1e^{-4}$ and a 1 cycle schedule [37].

4.2. Quantitative Evaluation

Datasets. We evaluate VL performance of our method on both indoor and outdoor datasets. For *indoor* scenes, we adopt the Indoor-6 dataset [9], which is a relatively large indoor dataset [9]. It comprises calibrated images from 6 multi-rooms indoor scenes captured over several days, leading to challenging VL. For *outdoor* scenes, the Map-Free [1] dataset is employed. We use its first 10 scenes, $s00000 \sim s00009$, following [4]. Each scene contains images from two scans of an outdoor location. We split them into mapping and query sets. The ground truth camera poses are computed via SfM method [34].

Indoor Visual Localization. Firstly, the indoor VL experiments are conducted using the Indoor-6 dataset. In the experiments, we compare our DIMM method and its variation enhanced by RANSAC (DIMM-R) with both APR [15]

Scene	SCR		APR		CRR	
(°/cm/%)	DSAC*[3]	ACE[4]	$marepo$[6]	$marepo_S$[6]	DIMM	DIMM-R
Throughput (fps)	17.9	17.9	55.6	55.6	**57.7**	20.1
s00000	1.2/5.1/55.8	0.7/4.2/62.6	**0.6**/3.6/69.2	**0.6**/3.2/70.7	0.7/**2.0**/93.1	**0.6/2.0/94.0**
s00001	0.8/2.4/94.5	**0.4/1.5/100**	0.9/1.8/96.0	0.8/1.6/99.2	1.1/1.6/96.8	1.1/1.6/96.8
s00002	0.8/2.4/69.5	**0.5/2.1**/77.7	0.6/2.3/77.1	**0.5/2.1/78.2**	0.9/3.7/71.2	0.9/3.0/77.1
s00003	0.6/2.1/75.8	**0.4**/2.4/73.6	0.9/2.5/80.3	0.8/**2.1**/84.5	0.6/2.4/92.8	0.6/2.2/**93.3**
s00004	1.1/6.3/40.4	**0.7**/5.8/45.3	0.8/5.3/45.9	**0.7**/5.1/47.6	**0.7**/2.4/82.6	**0.7/2.3/96.2**
s00005	0.7/3.6/64.4	**0.5**/2.4/73.3	0.7/2.7/72.2	**0.5**/2.6/77.3	0.8/2.5/86.2	0.8/**2.1/87.9**
s00006	1.0/4.4/53.4	0.8/4.5/54.8	0.8/4.2/58.5	**0.6/3.6**/63.8	0.8/5.6/44.3	0.8/3.9/**64.8**
s00007	0.7/7.8/16.9	0.6/7.6/19.0	1.0/6.3/25.1	0.8/6.1/27.5	**0.6/3.6**/69.5	**0.5/3.5/71.1**
s00008	1.2/3.5/68.9	**0.7**/3.1/71.6	**0.7**/2.3/77.4	**0.7/1.9**/82.8	**0.7**/2.7/85.1	**0.7**/2.4/**87.9**
s00009	0.8/3.1/81.2	**0.4/1.2/99.7**	0.8/1.8/94.9	0.9/1.6/95.1	0.7/1.3/97.7	0.7/**1.2**/97.4
Average	0.8/4.1/62.1	**0.6**/3.5/67.7	0.8/3.3/69.7	0.7/3.0/72.7	0.8/2.8/81.9	0.7/**2.4/86.6**

Table 3. **Pose accuracy comparison on the outdoor MapFree [1] dataset.** We report the median errors in degree (°) for the orientation, cm for the position, and recall at 5cm/5°. The **best** results, the second best and our results are highlighted.

Recall-5°/5cm (%)	s00000	s00001	s00002	s00003	s00004	s00005	Average
ACE [4]	62.6	100	**77.7**	73.6	45.3	73.3	72.1
DIMM-CRR (Ours)	**93.1**	96.8	71.2	**92.8**	**82.6**	**86.2**	**87.1**
DIMM-SCR	64.8	**98.6**	74.5	77.3	48.2	72.1	72.6
DIMM-APR	3.2	12.4	5.7	3.6	0.3	2.6	4.6

Table 4. **Ablation Results about Camera Ray Regression (CRR).** We evaluate three variations of our model to showcase the effectiveness of CRR, by modifying output to different VL formulations.

Feature Encoder	Med. R Err. (°)	Med. t Err. (cm)	Recall-5°/5cm
$\mathcal{T}_i \otimes \mathcal{T}_{[\mathsf{CLS}]} \otimes \mathcal{T}_{\mathsf{xy}_i}$	**0.8**	**5.6**	**44.3**
$\mathcal{T}_i^* \otimes \mathcal{T}_{[\mathsf{CLS}]}^* \otimes \mathcal{T}_{\mathsf{xy}_i}$	**0.8**	6.0	39.7
$\mathcal{T}_i \otimes \mathcal{T}_G$ [41] $\otimes \mathcal{T}_{\mathsf{xy}_i}$	1.1	8.3	19.7
$\mathcal{T}_i \otimes \mathcal{T}_{[\mathsf{CLS}]} \otimes \mathsf{XY}_i$	0.9	7.6	27.2
$\mathcal{T}_i \otimes \mathcal{T}_{[\mathsf{CLS}]}$	0.9	7.4	26.8
$\mathcal{T}_i$	1.1	11.5	17.2

Table 5. **Ablation Study about our Feature Encoder.** The experiments are performed on the MapFree s00006 dataset.

and SCR [3–5, 9, 20] approaches. The results are reported in Tab. 2. Additionally, we analyzed the corresponding mapping time and memory costs of various VL methods in Tab. 1, for a comprehensive comparison. DIMM demonstrates better or comparable performance than the previous *state-of-the-art*, *e.g.*, achieving a recall@5°/5cm of 78.3 in scene6. Moreover, our method achieves generally better results than EGFS-q with "hard" multi-mapper combination, proving the efficacy of our "soft" mapper ensemble utilizing semantic guidance. The effectiveness of RC-RANSAC is also highlighted by a substantial precision improvement in scene2a from 66.7 to 77.4. Overall, the proposed DIMM-R achieves the best average recall in this dataset. Furthermore, our method enables mapping with reduced computational costs compared to D2S and is trained without 3D supervision, as illustrated in Tab. 1. These findings underscore the potential of our CRR-based method as a promising VL implementation. Some visualization results of DIMM can be found in Fig. 4 and Fig. 7 in the Suppl., where the 3D scene structure is preserved, demonstrating the privacy protection offered by our method.

Outdoor Visual Localization. Our outdoor VL experiments are conducted on the Map-Free Dataset. In addition to the SCR-based ACE and DSAC*, we also compared the recent APR-based method, $marepo$ [6], and its fine-tuned variation, $marepo_S$. The results, including median errors for rotation (°) and translation (cm), as well as localization recalls at 5cm/5°, are presented in Tab. 3. These results also come from methods with mapping time in Tab. 1. From the table, it is evident that our methods achieve the best overall results, outperforming both SCR and APR methods. The rotation accuracy of ACE, yet, surpasses our methods, indicating that local ambiguity poses a significant challenge for CRR methods and obtaining precise ray directions from image patches is challenging. However, it is observed that CRR methods facilitate better camera center estimation compared to APR and SCR methods. This improvement can be attributed to the fact that the camera center location integrates contributions from all camera rays, and the bounded ray error [27] enhances the robustness of the center estimation against noise. The efficacy of *RC-RANSAC* algorithm can also be verified in the table, as it leads to substantial improvements of VL performance, *e.g.*, $44.3 \rightarrow 64.8$ of recall in s00006. Furthermore, we investigate the computational efficiency of our methods. The throughputs (fps) of all methods are reported in the table, revealing that the lightweight network and linear pose solver of DIMM achieve the best inference efficiency.

4.3. Ablation Experiments

Camera Ray Regression. One of our main contributions in this work is introducing Camera Ray Regression (CRR) into the VL task. Thus, we provide ablation study about this formulation in Tab. 4. In particular, we modify the last layer of DIMM to output scene coordinates (DIMM-SCR) or direct camera pose (DIMM-APR), and train these variations separately. The DIMM-SCR is trained with GDT, same as ACE[4]. The DIMM-APR, on the other hand, is trained in image level, as the direct pose vectors cannot

Figure 5. **Visualization results of sub-mapper attention score.** In the experiments, we observed that specific semantic entities elicit responses from different mappers. For instance, the fireplace at the top of the figure frequently results in high attention scores from mapper #3, while mapper #2 exhibits a low attention score. Conversely, mapper #3 seems do not "care" the sofa at the bottom, which leads to high scores of mapper #2 instead.

be assigned to image patches. The experiments are performed on the MapFree dataset, with the localization recall reported. In Tab. 4, we can see that our CRR formulation significantly outperforms the SCR and APR under the same network architecture, indicating the contribution of our ray-parameterization. DIMM-SCR overcomes ACE slightly in average, but our model is tailored for CRR. DIMM-APR utilizes per-scene pose regression and achieves poor results, but this is consistent with the results in $marepo$ [6]. This suggests the APR model may require much larger datasets to learn pose patterns. Overall, the results in Tab. 4 evaluate our main contribution: CRR formulation in VL.

Feature Encoder. Based on our analysis of CRR, we require patch features to exhibit strong local description while also incorporating global perception and image position encoding to achieve disambiguation. To validate this, we conduct experiments on the MapFree $s00006$ dataset (cf. Tab. 5). Specifically, we modify the content of the patch feature, including features w/o global perception and image position encoding ($\mathcal{T}_i \otimes \mathcal{T}_{[\mathsf{CLS}]}$ and $\mathcal{T}_i$), to compare with the proposed version ($\mathcal{T}_i \otimes \mathcal{T}_{[\mathsf{CLS}]} \otimes \mathcal{T}_{\mathsf{xy}_i}$). Also, we investigate the efficacy of DINO fine-tuning, and the performance of original DINO ($\mathcal{T}_i^* \otimes \mathcal{T}_{[\mathsf{CLS}]}^* \otimes \mathcal{T}_{\mathsf{xy}_i}$) is also reported. It is evident that global perception with image position encoding are crucial for CRR, leading to a significant accuracy improvement. The fine-tuning of DINO also improves the performance as well (39.7 $vs.$ 44.3). Visualization comparison about the global perception is presented in Fig. 4. We can see the proposed global perception enables the learning of ray geometry, causing the rays to converge toward the camera center, thereby enhancing precision. We also conduct experiments for a variation without Fourier Encoding ($\mathcal{T}_i \otimes \mathcal{T}_{[\mathsf{CLS}]} \otimes \mathsf{XY}_i$). The results in the first and third rows of Tab. 5 show that Fourier Encoding effectively prevents the submergence of positional

Atten. Num.	Med. R Err. (°)	Med. t Err. (cm)	Recall@5°/5cm	size (MB)
0 [1]	1.0	7.2	31.0	**12.1**
2	0.9	5.8	42.6	14.2
4	**0.8**	**5.6**	**44.3**	16.2
6	**0.8**	6.3	37.7	18.2
8	1.1	7.7	30.4	20.2

[1] Take average of the sub-mapper embeddings, eliminating attention layers.

Table 6. **Ablation Study about the Number of Semantic Attention Layers.** The experiments are performed on the MapFree $s00006$ dataset. The efficacy of our semantic attention is evaluated.

features. Finally, we explore another possible global perception way provided by [41] ($\mathcal{T}_i \otimes \mathcal{T}_G \otimes \mathcal{T}_{\mathsf{xy}_i}$), the feature for image retrieval. The results show that the replaced global feature fails to match the original efficacy, possibly because DINO's global feature is more compatible with its own local patch features, facilitating subsequent mapping learning.

Mapper Head. In the same scene, we also conduct ablation experiments on the network structure of the Mapper Head. We first validate the efficacy of the semantic attention module and experiment with its number of layers. When the layer number is set to 0, we directly take the average of output from multiple sub-mappers. The results are presented in Tab. 6, containing median rotation/translation error, localization recall and model size. It is evident that utilizing semantic attention is more effective compared to simple average fusion, and the increase in network size is deemed acceptable. Additional ablation study about the MLP mappers can be found in Suppl. B. To showcase the effectiveness of the soft mapper assemble, we visualize some query images combined with their sub-mapper attention scores in Fig. 5. It can be observed that different sub-mappers get high attention scores to different objects in images. This indicates that our semantic attention mechanism allows the sub-mappers to "remember" specific semantic objects.

5. Conclusion

In this work, we propose regressing camera rays to perform visual localization. This overparameterized representation of camera model leads to high precision, and, more importantly, enhances privacy protection. In particular, we introduce DIMM as a learning model to regress camera rays from images. It utilizes DINO as a scene-agnostic encoder to output features that incorporate both local and global perception. A scene-specific mapper then regresses ray parameters from the features, involving a semantic attention module to merge results from multiple mappers. Finally, a ray-level RANSAC algorithm is proposed to improve the ray-to-pose accuracy. In experiments on both indoor and outdoor datasets, our methods demonstrate comparable or superior performance to current approaches, while ensuring privacy preservation.
Acknowledgement. This work has been funded in part by the NSFC grants 62176156.

References

[1] Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Áron Monszpart, Victor Adrian Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *ECCV*, 2022. 6, 7

[2] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7525–7534, 2019. 1

[3] Eric Brachmann and Carsten Rother. Visual camera relocalization from rgb and rgb-d images using dsac. *IEEE transactions on pattern analysis and machine intelligence*, 44 (9):5847–5865, 2021. 3, 6, 7

[4] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *CVPR*, 2023. 1, 2, 3, 4, 5, 6, 7

[5] Bach-Thuan Bui, Huy-Hoang Bui, Dinh-Tuan Tran, and Joo-Ho Lee. D2s: Representing sparse descriptors and 3d coordinates for camera relocalization. *IEEE Robotics and Automation Letters*, 2024. 1, 2, 3, 6, 7

[6] 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*, pages 20665–20674, 2024. 1, 3, 7, 8

[7] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 6

[8] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*. 2

[9] Tien Do, Ondrej Miksik, Joseph DeGol, Hyun Soo Park, and Sudipta N Sinha. Learning to detect scene landmarks for camera localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11132–11142, 2022. 6, 7

[10] 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. 1, 3

[11] Kartik Garg, Sai Shubodh Puligilla, Shishir Kolathaya, Madhava Krishna, and Sourav Garg. Revisit anything: Visual place recognition via image segment retrieval. In *European Conference on Computer Vision*, pages 326–343. Springer, 2025. 1, 3

[12] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1, 3, 4, 5

[13] Sergio Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17658–17668, 2023. 4, 6

[14] Xudong Jiang, Fangjinhua Wang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. R-score: Revisiting scene coordinate regression for robust large-scale visual localization. *arXiv preprint arXiv:2501.01421*, 2025. 3, 4, 5

[15] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *2016 IEEE international conference on Robotics and Automation (ICRA)*, pages 4762–4769. IEEE, 2016. 1, 3, 6, 13

[16] 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*, pages 5974–5983, 2017.

[17] 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*, pages 2938–2946, 2015. 3

[18] Sihang Li, Siqi Tan, Bowen Chang, Jing Zhang, Chen Feng, and Yiming Li. Unleashing the power of data synthesis in visual localization. *arXiv preprint arXiv:2412.00138*, 2024. 13

[19] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023. 3

[20] Ting-Ru Liu, Hsuan-Kung Yang, Jou-Min Liu, Chun-Wei Huang, Tsung-Chih Chiang, Quan Kong, Norimasa Kobori, and Chun-Yi Lee. Reprojection errors as prompts for efficient scene coordinate regression. In *European Conference on Computer Vision*, pages 286–302. Springer, 2025. 1, 2, 3, 6, 7

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019. 6

[22] Deben Lu, Wendong Xiao, Teng Ran, Liang Yuan, Kai Lv, and Jianbo Zhang. Attention-based accelerated coordinate encoding network for visual relocalization. In *2024 IEEE 7th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 1675–1680, 2024. 1, 3, 4, 5

[23] Yuanxun Lu, Jingyang Zhang, Tian Fang, Jean-Daniel Nahmias, Yanghai Tsin, Long Quan, Xun Cao, Yao Yao, and Shiwei Li. Matrix3d: Large photogrammetry model all-in-one. *arXiv preprint arXiv:2502.07685*, 2025. 2, 3

[24] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. 2, 4

[25] Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. *arXiv preprint arXiv:2412.12392*, 2024. 3

[26] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 4

[27] Linfei Pan, Daniel Barath, Marc Pollefeys, and Johannes Lutz Schönberger. Global Structure-from-Motion Revisited. In *European Conference on Computer Vision (ECCV)*, 2024. 2, 3, 7

[28] Maxime Pietrantoni, Martin Humenberger, Torsten Sattler, and Gabriela Csurka. Segloc: Learning segmentation-based representations for privacy-preserving visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15380–15391, 2023. 1, 3

[29] Francesco Pittaluga, Sanjeev J. Koppal, Sing Bing Kang, and Sudipta N. Sinha. Revealing scenes by inverting structure from motion reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[30] 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*, pages 12716–12725, 2019. 1, 2, 6

[31] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. Back to the feature: Learning robust camera localization from pixels to pose. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3247–3257, 2021. 1

[32] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I 12*, pages 752–765. Springer, 2012. 2

[33] 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. 2

[34] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 6

[35] Thomas Schops, Viktor Larsson, Marc Pollefeys, and Torsten Sattler. Why having 10,000 parameters in your camera model is better than twelve. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2535–2544, 2020. 1, 2, 3

[36] 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*, pages 2733–2742, 2021. 3

[37] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of residual networks using large learning rates. *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, 2019. 6

[38] Pablo Speciale, Johannes L. Schönberger, Sing Bing Kang, Sudipta N. Sinha, and Marc Pollefeys. Privacy preserving image-based localization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5488–5498, 2019. 1

[39] 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*, pages 8922–8931, 2021. 1, 3

[40] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020. 4

[41] 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. 1, 2, 3, 4, 7, 8

[42] Yifan Wang, Xingyi He, Sida Peng, Dongli Tan, and Xiaowei Zhou. Efficient loftr: Semi-dense local feature matching with sparse-like speed. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21666–21675, 2024. 3

[43] Yinshuang Xu, Dian Chen, Katherine Liu, Sergey Zakharov, Rares Andrei Ambrus, Kostas Daniilidis, and Vitor Campagnolo Guizilini. $se(3)$ equivariant ray embeddings for implicit multi-view depth estimation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 3

[44] Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. In *International Conference on Learning Representations (ICLR)*, 2024. 2, 3, 4

[45] Yesheng Zhang and Xu Zhao. Mesa: Matching everything by segmenting anything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20217–20226, 2024. 3