Adversarial Exploitation of Data Diversity Improves Visual Localization

Sihang Li* Siqi Tan* Bowen Chang Jing Zhang Chen Feng✉ Yiming Li✉
New York University
{sihangli, tan.k, cfeng, yimingli}@nyu.edu
https://ai4ce.github.io/RAP

Abstract

Visual localization, which estimates a camera's pose within a known scene, is a fundamental capability for autonomous systems. While absolute pose regression (APR) methods have shown promise for efficient inference, they often struggle with generalization. Recent approaches attempt to address this through data augmentation with varied viewpoints, yet they overlook a critical factor: appearance diversity. In this work, we identify appearance variation as the key to robust localization. Specifically, we first lift real 2D images into 3D Gaussian Splats with varying appearance and deblurring ability, enabling the synthesis of diverse training data that varies not just in poses but also in environmental conditions such as lighting and weather. To fully unleash the potential of the appearance-diverse data, we build a two-branch joint training pipeline with an adversarial discriminator to bridge the syn-to-real gap. Extensive experiments demonstrate that our approach significantly outperforms state-of-the-art methods, reducing translation and rotation errors by 50% and 41% on indoor datasets, and 38% and 44% on outdoor datasets. Most notably, our method shows remarkable robustness in dynamic driving scenarios under varying weather conditions and in day-to-night scenarios, where previous APR methods fail.

1. Introduction

Visual localization, the task of calculating a 6-DoF camera pose—its translation and rotation—based on a query image within a given environment, is essential for various applications, including robotics [2], autonomous vehicles [20], and virtual reality [13]. Besides traditional geometry-based approaches, recent learning-based visual localization methods adopt absolute pose regression (APR) [7, 10, 26, 58], scene coordinate regression (SCR) [5, 6, 46, 66], or post pose refinement (PPR) [11, 21, 35, 46, 64, 71, 73]. SCR methods focus on learning-based 2D-3D correspondences followed by subsequent Perspective-n-Point (PnP) for pose estimation. PPR methods heavily rely on a pose prior, usually obtained from image retrieval, followed by iterative refinement. In contrast, APR methods employ a supervised framework to train a regression neural network on image-pose pairs, enabling direct pose estimation during inference. APR offers faster runtime and lower error in challenging scenes with sparse views, significant lighting variations, or numerous dynamic objects, making it a promising method for ensuring robustness in real-world applications.

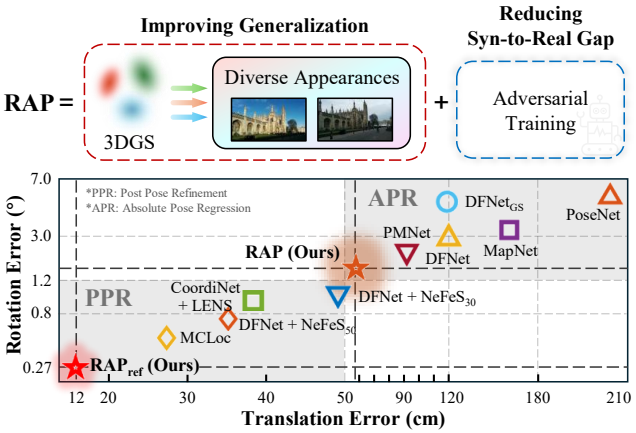

Figure 1. We propose RAP, a novel pipeline to train robust APR models. We lift real-world 2D images into 3D Gaussian Splats [27] to synthesize images with *diverse appearances and poses*, improving model generalizability. We also introduce an adversarial discriminator, mitigating the syn-to-real gap to learn robust features. Together, we achieve state-of-the-art performance.

Despite promises, there is a performance gap in localization accuracy between APR and other methods. A well-known pivotal work [54] attributes this to APR performing image-based memorization, i.e., retrieving poses seen during training. Driven by this crucial finding, to improve such memorization while avoiding the need for denser real-world training samples, recent methods leverage Neural Radiance Fields (NeRF) [42] to synthesize additional posed images for APR training [10, 32, 45]. LENS [45] tried to employ appearance perturbation using NeRF-W [41], but found the

*Equal contribution. ✉Corresponding author.

improvements minimal [45]. As appearance augmentation is common in other learning tasks, why does it fail in APR?

We hypothesize a learning gap: Limitations in previous training pipelines prevented the effective use of diverse data to boost performance. Artifacts always exist in images rendered by common novel view synthesis (NVS) methods, which might disturb the model feature space. Inspired by generative adversarial networks (GAN) [22], where a discriminator is trained to distinguish between real and generated samples, we propose adversarial training for APR, designing a discriminator to align the features of synthetic and real images, thereby reducing the syn-to-real domain gap and mitigating the impact of rendering artifacts. Augmentation quality also matters. To efficiently synthesize diverse high-quality images with *controllable* varying appearance, we extend the vanilla 3D Gaussian Splatting (3DGS) [27] to appearance-varying 3DGS with deblurring ability.

These form our two-branch joint training framework for **r**obust **a**bsolute **p**ose regression (**RAP**). The first branch coarsely trains our Transformer-based pose regressor with both real data and data synthesized at the original real pose, together with an adversarial discriminator to reduce the syn-to-real domain gap. The second branch progressively generates randomly perturbed poses and appearances, providing additional supervision to the same APR Transformer. Through extensive experiments, we demonstrate that exploiting data diversity using adversarial training significantly increases localization accuracy in APR. Meanwhile, our results indicate that APR consistently benefits from more diverse visual data, and we observe clear signs of a more generalizable APR emerging with its localization performance cannot be explained merely by memorization.

Our contributions are summarized as follows:

- We identify the crucial role of appearance diversity for APR, and develop a 3DGS-based appearance-varying data augmentation framework to efficiently generate diverse synthetic data with controllable lighting conditions.
- We propose an adversarial discriminator to reduce the syn-to-real gap. Together with progressive data synthesis, we form a robust two-branch joint training pipeline that fully unleashes the power of data diversity.
- We conduct extensive experiments showing our method outperforms state-of-the-art approaches on challenging datasets with significant appearance change. Ablation studies further analyze key factors affecting performance.

2. Related Works

Visual Localization. Visual localization aims to estimate a camera's translation and rotation within a 3D scene. Traditional geometry-based methods [8, 17, 34, 37, 48, 50–52, 62] accomplish this by using point clouds and a reference image database, relying on stored descriptors and image retrieval to establish 2D-3D correspondences. In contrast, scene coordinate regression (SCR) methods [4–6, 66] embed map information within neural networks to directly predict 2D-3D correspondences. Both approaches generally require PnP [19] and RANSAC [18] to output camera poses at test time, which adds additional computation cost. Alternatively, absolute pose regression (APR) [7, 9, 24, 26, 44, 57] aims to directly regress the camera pose from a query image using neural networks. Although the performance is suboptimal compared with geometry-based methods, APR remains a promising approach due to its fast inference.

Data Augmentation for Pose Regression. End-to-end pose regression methods rely heavily on the amount and diversity of training data. Previous work [54] shows that APR implicitly learns image *retrieval* in the given environment. Therefore, the following works LENS [45], DFNet [10] and PMNet [32] enhance APR performance by spatially enriching training views with NeRF. However, these approaches fail to address the generalizability of APR models and exhibit several limitations: (1) The efficiency of training and novel view synthesis (NVS) in NeRF is severely restricted, hindering scalability. (2) They limit NVS to geometric (pose) transformations while neglecting photometric (appearance) variations, thereby decreasing APR robustness to changes in visual appearance. (3) The augmented data is underutilized in their learning frameworks, leaving its potential for improving APR largely untapped. Differently, our framework switches to 3DGS [27] as the scene representation to efficiently generate novel posed images with *controllable* appearances and introduce adversarial training to unleash the power of such diverse data.

Handling and Synthesizing Challenging Scenarios. Visual localization often encounters unstructured photo collections [61], where visual appearance varies due to moving objects, lighting changes, and inconsistent camera exposure settings. To tackle these in-the-wild challenges, NeRF-W [40] uses per-image transient and appearance embeddings. In 3DGS [27], VastGaussian [33] applies a CNN to 3DGS outputs but still struggles with significant appearance variations. SWAG [14] mitigates this issue by storing appearance information in an external hash-grid-based implicit field, while GS-W [70] enhances flexibility by separating intrinsic and dynamic appearance features for each Gaussian point. 3DGM [31] leverages consensus across multiple sequences as the self-supervision signal to remove transient and moving objects without human annotations. Deblur-GS [68] addresses motion blur—another challenge in localization datasets—by modeling camera motion to yield sharper edges in rendered scenes. Our method incorporates appearance modeling and edge refinement to handle and synthesize diverse indoor, outdoor, and driving scenes.

3. Method

3.1. Pre-Processing with 3DGS

A robust pose regressor should focus on intrinsic scene attributes, not appearance variations. Therefore, we first

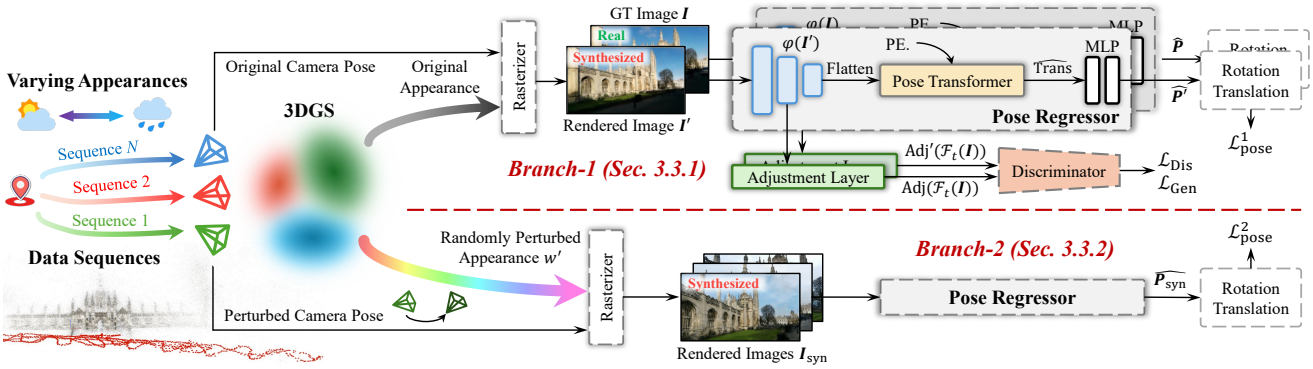

Figure 2. **Pipeline of RAP.** We lift multiple RGB video sequences into 3D Gaussian Splats, which serve as our data engine. The **branch-1** (see Sec. 3.3.1) inputs paired real and synthetic images to regress poses, with a discriminator to bridge the syn-to-real gap. The **branch-2** (see Sec. 3.3.2) generates views with novel poses and appearances, which are fed into the same pose regressor as additional supervision.

synthesize diverse visual data for training. We leverage 3DGS [27], representing scenes with explicit ellipsoids, to model diverse appearances. Following GS-W [70], we assume the scene contains K Gaussians and represent the independent intrinsic material attributes using positions $\boldsymbol{\mu} \in \mathbb{R}^{K\times 3}$, spherical harmonics $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{K\times 16\times 3}$, and other parameters $\boldsymbol{\Theta}$ including rotation $\boldsymbol{q} \in \mathbb{R}^{K\times 4}$, scaling $\boldsymbol{s} \in \mathbb{R}^{K\times 3}$, and opacity $\boldsymbol{\alpha} \in \mathbb{R}^{K}$. To capture the dynamic appearance influenced by environmental factors, we extract features from the input image and assign each Gaussian its own feature using a learnable sampler $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{K\times 2}$, forming features $\boldsymbol{\mathcal{E}} \in \mathbb{R}^{K\times 16\times 3}$. We also incorporate the camera's view direction θ to account for viewpoint-dependent effects. The final color of Gaussians $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{K\times 3}$ is:

$$\boldsymbol{\mathcal{C}} = \mathrm{MLP}(\boldsymbol{\mu}, \boldsymbol{\mathcal{Y}}, \omega\boldsymbol{\mathcal{E}}, \theta), \tag{1}$$

where ω is the blending weight that controls the dynamic appearance of the rendered image.

Another significant challenge in visual localization is motion blur, often caused by slow shutter speeds during video capture, leading to pose ambiguity and degraded rendering quality, further decreasing localization accuracy. Inspired by Deblur-GS [68], we model camera motion blur as the inverse of scene motion, i.e., the transformation in Gaussian position denoted by $\boldsymbol{\mathcal{T}} \in \mathrm{SE}(3)$. For each training image, we sample a certain time step along a linear trajectory with a sampling weight $\phi \in \mathbb{R}^n$ and blend them to compute loss $\mathcal{L}$ with the original blurry image $\boldsymbol{\mathcal{I}}_b \in \mathbb{R}^{H\times W\times 3}$, optimizing $\boldsymbol{\mathcal{T}}$, ϕ, $\boldsymbol{\mathcal{C}}$ and other 3DGS parameters $\boldsymbol{\Theta}$:

$$\underset{\phi,\boldsymbol{\mathcal{T}},\boldsymbol{\mu},\boldsymbol{\mathcal{Y}},\boldsymbol{\mathcal{E}},\boldsymbol{\Theta}}{\operatorname{argmin}} \mathcal{L}\left(\boldsymbol{\mathcal{I}}_b, \sum_{i=1}^{n} \phi_i \,\mathrm{Render}(\boldsymbol{\mathcal{T}}_i(\boldsymbol{\mu}), \boldsymbol{\mathcal{C}}, \boldsymbol{\Theta})\right), \tag{2}$$

where the details of $\mathcal{L}$ are in supplemental materials. After training, our 3DGS can efficiently render posed images given θ and ω.

3.2. Architecture of Pose Regressor

Given a set of images and their associated camera poses $\{(\boldsymbol{I}_i, \boldsymbol{P}_i)\}_{i=1}^n$, our goal is to train a neural network to directly output a homogeneous camera pose $\boldsymbol{P} \in \mathbb{R}^{3\times 4}$ for a query image $\boldsymbol{I} \in \mathbb{R}^{H\times W\times C}$. Our network architecture is shown as the pose regressor in Fig. 2.

Feature Extractor. Pose regression networks typically extract features using a common backbone φ, such as VGG [60] or EfficientNet [63], leveraging multiple deeper layers for translation and rotation regression:

$$\varphi(\boldsymbol{I}) = \{\mathcal{F}_0(\boldsymbol{I}), ..., \mathcal{F}_{N-1}(\boldsymbol{I}), \mathcal{F}_N(\boldsymbol{I})\}, \tag{3}$$

$\mathcal{F}_*(\cdot)$ denotes features extracted from the $*$-th layer of a backbone with N layers. $\mathcal{F}_t(\boldsymbol{I})$ and $\mathcal{F}_r(\boldsymbol{I})$ denote features for translation and rotation regression, respectively.

Pose Transformer. Unlike CNN-based regression models [10, 32], where fine-grained local features can introduce noise and harm performance, we propose *Pose Transformer* to leverage the strong ability of Vision Transformer (ViT) [16] for modeling long-range dependencies. Each Transformer generates a global token (`Trans` for translation and `Rot` for rotation) to provide a comprehensive context for pose regression, inspired by the `CLS` token in ViT. Given $\mathcal{F}_r(\boldsymbol{I})$ and $\mathcal{F}_t(\boldsymbol{I})$, the translation token is then concatenated with the flattened input features*:

$$\widetilde{\mathcal{F}_t}(\boldsymbol{I}) = \mathrm{Cat}(\mathrm{Flatten}(\mathcal{F}_t(\boldsymbol{I})), \texttt{Trans}) \in \mathbb{R}^{(H_t W_t+1)\times C_t}. \tag{4}$$

The positional encodings are then added to the flattened feature $(\texttt{PE} + \widetilde{\mathcal{F}_t}(\boldsymbol{I})) \in \mathbb{R}^{(H_t W_t+1)\times C_t}$. Multi-head Self-Attention (MSA) is then conducted through a stack of multiple layers with the post-processing as follows:

$$\begin{aligned} \widehat{\mathcal{F}_t'}(\boldsymbol{I}) &= \mathrm{MSA}(\texttt{PE} + \widetilde{\mathcal{F}_t}(\boldsymbol{I})) + \texttt{PE} + \widetilde{\mathcal{F}_t}(\boldsymbol{I}), \\ \widehat{\mathcal{F}_t}(\boldsymbol{I}) &= \mathrm{LN}(\mathrm{FFN}(\mathrm{LN}(\widehat{\mathcal{F}_t'}(\boldsymbol{I}))) + \widehat{\mathcal{F}_t'}(\boldsymbol{I})), \end{aligned} \tag{5}$$

*We only present the translation regression for simplicity.

where LN indicates layer normalization and FFN denotes the fully connected feed-forward network, consisting of two linear layers with a ReLU. The final output is flattened back to $(H_t W_t + 1) \times c_t$. See supplementary for more details.

Regression Head. Only the processed translation token, $\widehat{\texttt{Trans}}$, capturing global features for regression, is fed into the regression head. This regression head consists of two MLPs, each with a hidden layer and GeLU activation:

$$\hat{\boldsymbol{t}} = \text{Linear}(\text{GeLU}(\text{Linear}(\widehat{\texttt{Trans}}))). \tag{6}$$

The $\hat{\boldsymbol{t}}$ represents the final prediction for translation. Similarly, we obtain the rotation prediction denoted by $\hat{\boldsymbol{r}}$.

3.3. Two-Branch Joint Training Paradigm

3.3.1. Branch-1: Aligning Features via Discriminator

Synthetic images from 3DGS provide novel viewpoints and appearances but often contain artifacts, leading to a syn-to-real domain gap. To align features from rendered and real images of the same pose, we introduce an adversarial training mechanism besides the basic pose regression training.

Pose Regression Loss. For basic training, we render the synthetic image $\boldsymbol{I'}$ with the same pose label $\boldsymbol{P}$ as the real image $\boldsymbol{I}$, both used as supervision for the pose regressor. The training objective consists of translation loss $\mathcal{L}_t$ and the rotation loss $\mathcal{L}_r$, which are measured by the Euclidean distance between the ground truth pose $\boldsymbol{P} = \{\boldsymbol{t}, \boldsymbol{r}\}$ and the estimated pose $\widehat{\boldsymbol{P}} = \{\hat{\boldsymbol{t}}, \hat{\boldsymbol{r}}\}$:

$$\mathcal{L}_t = \|\boldsymbol{t} - \hat{\boldsymbol{t}}\|_2, \tag{7}$$

$$\mathcal{L}_r = \left\| \boldsymbol{r} - \frac{\hat{\boldsymbol{r}}}{\|\hat{\boldsymbol{r}}\|} \right\|_2, \tag{8}$$

$$\mathcal{L}^1_{\text{pose}} = \mathcal{L}_t \exp(-s_t) + s_t + \mathcal{L}_r \exp(-s_r) + s_r, \tag{9}$$

where s_t and s_r are learned parameters for balancing the optimization between rotation and translation [25].

Adversarial Loss. The adversarial training mechanism optimizes the discriminator to distinguish real from rendered image features, while training the feature extractor to fool the discriminator, effectively bridging the domain gaps. To prevent vanishing gradients, we propose a novel adversarial objective for pose regression, inspired by LSGAN [39]:

$$\begin{aligned}\underset{D}{\text{argmin}}\, \mathcal{L}_{\text{Dis}}(D) =& \frac{1}{2}\mathbb{E}_{\boldsymbol{I}\sim p_{\text{data}}(\boldsymbol{I})}\left[(D(\text{Adj}(\mathcal{F}_t(\boldsymbol{I}))) - 1)^2\right] \\ &+ \frac{1}{2}\mathbb{E}_{\boldsymbol{I'}}\left[D(\text{Adj}'(\mathcal{F}_t(\boldsymbol{I'}))^2\right], \end{aligned} \tag{10}$$

$$\underset{G}{\text{argmin}}\, \mathcal{L}_{\text{Gen}}(G) = \frac{1}{2}\mathbb{E}_{\boldsymbol{I'}}\left[(D(\text{Adj}'(\mathcal{F}_t(\boldsymbol{I'})) - 1)^2\right]. \tag{11}$$

Here, Adj and Adj′ are the adjustment layers, consisting of Conv-ReLU-BN layers. The feature extractor φ acts as the generator G, while D is the discriminator, composed of several convolutional layers with ReLU activations. More details are in supplemental materials.

3.3.2. Branch-2: Training while Synthesizing Data

With the proposed appearance-varying 3DGS, more posed images are generated to enrich the training data for better generalizability. Specifically, our data synthesis is categorized into two dimensions: *pose augmentation* and *appearance augmentation*, as illustrated in Fig. 1. For pose augmentation, given a training pose $\boldsymbol{P}$, a perturbed pose $\boldsymbol{P}_{\text{syn}}$ is generated around $\boldsymbol{P}$ by the translation noise of δt and rotation noise of δr. For appearance augmentation, we randomly adjust the appearance of rendered images using random blending weights ω, and then render the synthetic image $\boldsymbol{I}_{\text{syn}}$ using the Gaussian Splats trained in Sec. 3.1. The novel image-pose pair $(\boldsymbol{I}_{\text{syn}}, \boldsymbol{P}_{\text{syn}})$, online generated every 20 epochs during training until the validation MSE loss and median errors cease to decrease, serves as additional supervision for the training. Given the estimated pose of the synthesized image denoted by $\widehat{\boldsymbol{P}_{\text{syn}}}$, the loss function $\mathcal{L}^2_{\text{pose}}(\widehat{\boldsymbol{P}_{\text{syn}}}, \boldsymbol{P}_{\text{syn}})$ is same as $\mathcal{L}^1_{\text{pose}}$.

3.3.3. Overall Objective

The total loss for the pose regressor is:

$$\mathcal{L}_{\text{total}} = \beta_1 \mathcal{L}^1_{\text{pose}} + \beta_2 \mathcal{L}^2_{\text{pose}} + \beta_3(\mathcal{L}_{\text{Gen}} + \mathcal{L}_{\text{Dis}}), \tag{12}$$

where $\beta_1, \beta_2, \beta_3$ are loss weights. The total loss will optimize the pose regressor, adjustment layers, and discriminator. Only the pose regressor will be deployed in the inference phase, while the other two components are discarded.

4. Experiments

4.1. Evaluation Setup

Datasets. We follow previous works [10, 32] to mainly use four scenes in the Cambridge Landmarks dataset [26] with spatial extents around 875 m^2. Moreover, we evaluate our method on MARS [30], a self-driving dataset featuring challenges like moving objects, lighting changes, and motion blur. To investigate the robustness of our model under extreme lighting changes, such as the transition from day to night, we also prepared a subset of the Aachen Day-Night dataset [53]. The training data includes images captured using various camera models with differing resolutions, which renders direct evaluation with APR methods infeasible. Thus, we standardized the camera models through center cropping and built a COLMAP [55] model as pose annotations, including 13 nighttime images for evaluation and 246 daytime images for training 3DGS and RAP. We also employ the 7-Scenes dataset [59], which provides seven indoor scenes with volumes spanning 1 m^3–18 m^3, and follow the original training and testing splits with more accurate SfM pose annotations [6, 11]. Although it is an indoor dataset, it is still non-trivial as it includes texture-less surfaces, object occlusions, and motion blur.

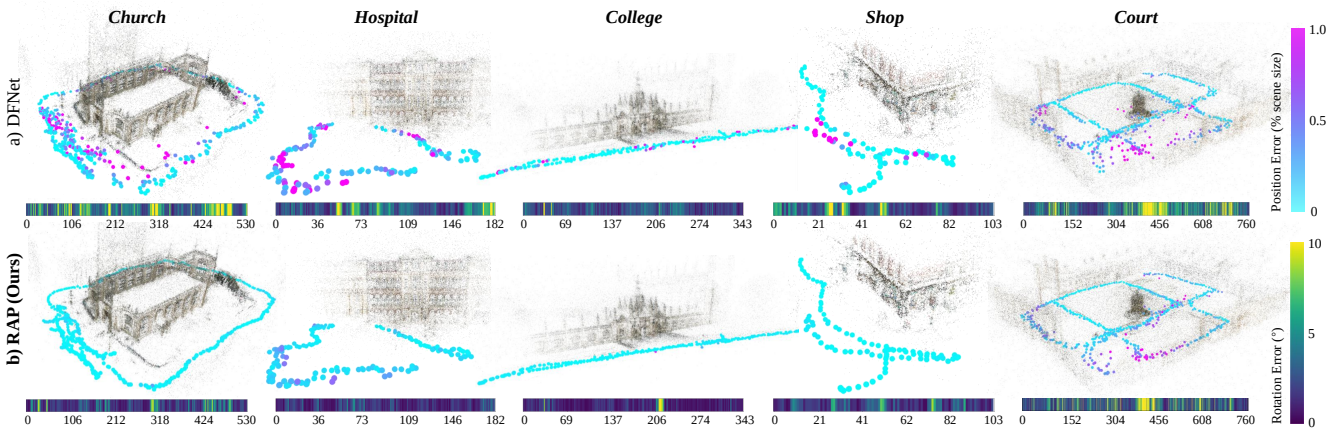

Figure 3. **Qualitative comparison of camera pose estimation errors between a) DFNet [10] and b) our RAP framework** across five scenes on the Cambridge Landmarks dataset [26]. Our RAP framework estimates trajectories that more closely follow the ground truth, with significantly reduced rotation and position errors compared to DFNet [10].

Table 1. **Median translation (cm) and rotation (°) errors on the Cambridge Landmarks dataset [26].** The best results† in pure APR and PPR are highlighted in **bold**.

	Methods	*College*	*Hospital*	*Shop*	*Church*	Average‡	*Court*
Pure APR	PN [26]	166/4.86	262/4.90	141/7.18	245/7.95	204/6.23	683/3.50
	MapNet [7]	107/1.89	194/3.91	149/4.22	200/4.53	163/3.64	N/A
	MS-Trans. [58]	83/1.47	181/2.39	86/3.07	162/3.99	128/2.73	N/A
	PAE [56]	90/1.49	207/2.58	99/3.88	164/4.16	140/3.03	N/A
	LENS† [45]	33/0.50	44/0.90	27/1.60	53/1.60	39/1.20	N/A
	DFNet [10]	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 (Ours)	**52/0.90**	**87/1.21**	**33/1.48**	**53/1.52**	**56/1.28**	**115/1.68**
SCR	DSAC* [4]	18/0.3	21/0.4	5/0.3	15/0.6	15/0.4	34/0.2
	ACE [6]	28/0.4	31/0.6	5/0.3	18/0.6	21/0.5	43/0.2
	GLACE [66]	19/0.3	17/0.4	4/0.2	9/0.3	12/0.3	19/0.1
PPR	FQN-MN [21]	28/0.38	54/0.82	13/0.63	58/2.00	38/0.96	4253/39.16
	LENS [45]	34/0.54	45/0.96	28/1.66	54/1.66	40/1.21	N/A
	CrossFire [46]	47/0.7	43/0.7	20/1.2	39/1.4	37/1.0	N/A
	$NeFeS_{50}$ [11]	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
	MCLoc [64]	31/0.42	39/0.73	12/0.45	26/0.88	27/0.62	N/A
	$DFNet_{GS\text{-}CPR}$ [36]	23/0.32	42/0.74	10/0.36	27/0.62	26/0.51	N/A
	$ACE_{GS\text{-}CPR}$ [36]	20/0.29	21/0.40	5/0.24	13/0.40	15/0.33	N/A
	$DFNet_{ref\ (Ours)}$	16/0.24	21/0.41	8/0.42	10/0.26	14/0.33	25/**0.13**
	RAP_{ref} (Ours)	**15/0.23**	**18/0.38**	**5/0.23**	**9/0.23**	**12/0.27**	**22**/0.15

‡Since most methods did not report results on *Court*, it is excluded from the average error calculation. †As CoordiNet + LENS [45] does not provide open-source code, it is unclear whether any post-processing is used.

Baselines. We first compare our proposed RAP against common APR-only approaches on the four datasets, where PMNet [32] and DFNet [10] are the most related and advanced methods based on data augmentation. We split the remaining methods into two categories based on whether they rely on extra novel view synthesis in test time, including SCR [5, 6, 66] and PPR (Post Pose Refinement) [11, 21, 35, 46, 64, 71, 73], which involves rendering images, querying features in novel views by the initial pose, iterative refinement or sequential refinement [44].

Implementation Details. First, we optimize our 3DGS for each scene without masking moving objects. We

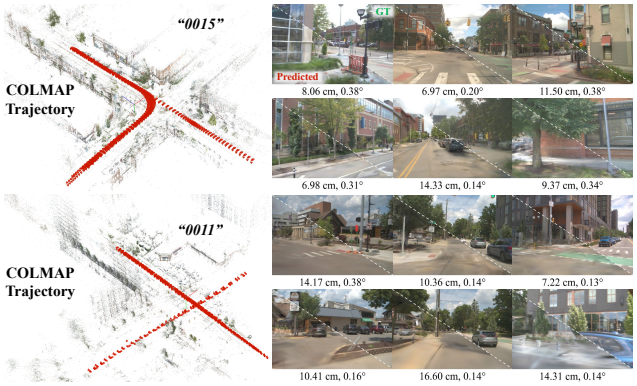

Figure 4. **Visualization of RAP_{ref} on MARS [30].** In each subfigure, a diagonal line separates the "Predicted" (rendered from the refined pose) and "GT" (ground truth) sections. Smooth alignment along this boundary shows RAP_{ref}'s improved pose accuracy.

Table 2. **Median translation (cm) and rotation (°) errors on the MARS dataset [30].**

Methods	*"0011"*	*"0015"*	*"0037"*	*"0041"*	Average
PoseNet [26]	149/1.80	136/2.34	123/1.60	75/0.92	121/1.67
RAP (Ours)	32/0.61	37/1.08	15/0.35	28/0.35	28/0.60
RAP_{ref} (Ours)	**8.5/0.13**	**8.2/0.20**	**8.7/0.09**	**7.6/0.11**	**8.3/0.13**

Table 3. **Median translation (COLMAP [72] unit) and rotation (°) errors on the Aachen Day-Night Dataset [53].**

APR-Based					SCR-Based	
PoseNet [26]	DFNet [10]	**RAP w/o App.**	**RAP**	**RAP_{ref}**	ACE [6]	GLACE [66]
217/74.30	174/85.80	134/75.99	130/13.70	**50/3.93**	914/90.50	482/36.4

then train our RAP network, which uses an EfficientB0 backbone [38] pre-trained on ImageNet [15], optimized with Adam [28] at a learning rate of 10^{-4}. Only the features from the third (`reduction_3`) and fourth (`reduction_4`) layers are used respectively for translation and rotation regression, and both layers are utilized for

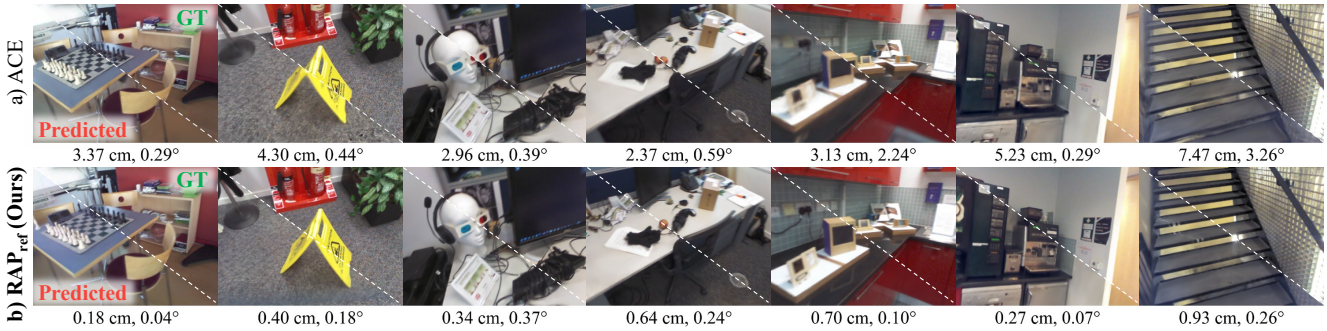

Figure 5. **Visualization of the localization errors of RAP_{ref} on the 7-Scenes dataset [59].**

Table 4. **Quantitative results on the 7-Scenes dataset [59].** The best results in pure APR and PPR are highlighted in **bold**. **DSLAM GT** and **SfM GT** refer to different sets of ground truth. More visualizations and details are in supplemental materials.

Category	Methods	*Chess*	*Fire*	*Heads*	*Office*	*Pumpkin*	*Kitchen*	*Stairs*	Average
Pure APR	PoseNet (PN) [26]	32/8.12	47/14.4	29/1.20	48/7.68	47/8.42	59/8.64	47/13.80	44/10.4
	MapNet [7]	8/3.25	27/11.7	18/13.3	17/51.5	22/4.02	23/4.93	30/12.1	21/7.77
	AtLoc+ [65]	10/3.18	26/10.8	14/11.4	17/5.16	20/3.94	16/4.90	29/10.2	19/7.08
	MS-Transformer [58]	11/4.66	24/9.60	14/12.2	17/5.66	18/4.44	17/5.94	17/5.94	18/7.28
	PAE [56]	12/4.95	24/9.31	14/12.5	19/5.79	18/4.89	18/6.19	25/8.74	19/7.48
	CoordiNet + LENS [45]	4/1.38	11/3.77	8/5.86	8/1.98	9/2.27	10/2.27	15/3.67	9/3.07
	DFNet [10]	5/1.88	17/6.45	6/**3.63**	8/2.48	10/2.78	22/5.45	16/3.29	12/3.71
	PMNet [32]	4/1.70	10/4.51	7/4.23	7/1.96	14/3.33	14/3.36	16/3.62	10/3.24
	RAP (Ours, DSLAM GT)	**3/1.41**	**7/3.46**	**6**/6.02	**5**/1.97	**6/1.96**	**7/2.18**	**10/2.14**	**6/2.73**
	RAP (Ours, SfM GT)	**2/0.85**	**6/2.84**	**4**/4.52	**4/1.57**	**3/1.10**	**5/1.10**	**10/1.30**	**5/1.90**
SCR	DSAC [4]	0.5/0.17	0.8/0.28	0.5/0.34	1.2/0.34	1.2/0.28	0.7/0.21	2.7/0.78	1.1/0.34
	ACE [6]	0.5/0.18	0.8/0.33	0.5/0.33	1.0/0.29	1.0/0.22	0.8/0.20	2.9/0.81	1.1/0.34
	GLACE [66]	0.6/0.18	0.9/0.34	0.6/0.34	1.1/0.29	0.9/0.23	0.8/0.20	3.2/0.93	1.2/0.36
	marepo‡ [12]	2.6/1.35	2.5/1.42	2.3/2.21	3.6/1.44	4.2/1.55	5.1/1.99	6.7/1.83	3.9/1.68
PPR	FQN-MN [21]	4.1/1.31	10.5/2.97	9.2/2.45	3.6/2.36	4.6/1.76	16.1/4.42	139.5/34.67	28/7.3
	CrossFire [46]	1/0.4	5/1.9	3/2.3	5/1.6	3/0.8	2/0.8	12/1.9	4.4/1.38
	DFNet + $NeFeS_{50}$ [11]	2/0.57	2/0.74	2/1.28	2/0.56	2/0.55	2/0.57	5/1.28	2.4/0.79
	HR-APR [35]	2/0.55	2/0.75	2/1.45	2/0.64	2/0.62	2/0.67	5/1.30	2.4/0.85
	MCLoc [64]	2/0.8	3/1.4	3/1.3	4/1.3	5/1.6	6/1.6	6/2.0	4.1/1.43
	DFNet + GS-CPR (SfM GT) [36]	0.7/0.20	0.9/0.32	0.6/0.36	1.2/0.32	1.3/0.31	0.9/0.25	2.2/0.61	1.1/0.34
	ACE + GS-CPR (SfM GT) [36]	0.5/0.15	0.6/0.25	0.4/0.28	0.9/0.26	1.0/0.23	0.7/0.17	1.4/0.42	0.8/0.25
	RAP_{ref} (Ours, DSLAM GT)	2.78/1.43	2.07/1.23	1.53/1.87	2.49/1.20	4.47/1.56	4.21/1.83	3.24/1.18	2.97/1.47
	RAP_{ref} (Ours, SfM GT)	**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**

‡As *marepo* [12] combines SCR and APR, we classify it as SCR.

narrowing the domain gap via the discriminator, which is also optimized with Adam [28], using a learning rate of 10^{-4} and betas set to (0.5, 0.999). More details about training are in supplemental materials. For generating random views, we apply random normalized perturbations to each training pose: $\delta t = 20$ cm and $\delta r = 10°$ for indoor scenes, and $\delta t = 150$ cm and $\delta r = 4°$ for outdoor scenes.

To allow for comparison with SCR methods and leverage 3DGS's efficient rendering for PPR, we extend the APR pipeline with match-based refinement similar to GS-CPR [36], denoted as RAP_{ref}. At test time, RAP's initial pose is used to render an RGB-D image via 3DGS. Together with MASt3R [29], we can obtain 2D-3D correspondences to perform RANSAC-PnP [18, 19], resulting in a refined pose. More details are in supplementary materials.

4.2. Benchmark Results

Cambridge Landmarks [26]. In the challenging outdoor Cambridge Landmarks dataset (Table 1), our RAP reduces both translation and rotation errors across all scenes by over 30% compared to other APR-only methods. The visualization in Fig. 3 shows that our method produces fewer outliers than DFNet [10]. In the three larger-scale scenes with significant appearance diversity (*College*, *Church*, and *Court*), rotation error is even halved compared to DFNet. Table 1 also shows the effectiveness of our RAP_{ref} in further reducing pose errors through refinement. RAP_{ref} outperforms CoordiNet + LENS [45], which assumes a continuous trajectory when an Extended Kalman Filter [23] is required for refinement [44]. RAP_{ref} even surpasses ACE [6] and its post-refinement variant, ACE + GS-CPR [36], despite GS-CPR manually masking dynamic objects when building 3DGS. This demonstrates the strong representation capability of our appearance-varying 3DGS with deblurring.

MARS [30]. Autonomous driving scenarios present unique challenges, including moving objects and frequent changes in lighting conditions, as illustrated in Fig. 4. Our RAP demonstrates effective and robust performance across four challenging scenes, as shown in Table 2, achieving an average of 28 cm / 0.60° localization error. This significantly

Table 5. **Ablation study.**

Setups on *Shop*	**Trans.** (cm) ↓	**Rot.** (°) ↓
I (Baseline): φ = VGG16	174	5.45
II: φ = Efficient-B0	103	4.64
III: II + Pose Aug.	75	3.52
IV: III + Appearance Aug.	60	3.14
V: IV + Decoder (ConvNet)	52	2.51
VI: V + Decoder (Transformer)	40	1.98
VII (Ours): VI + Discriminator	**33**	**1.48**

outperforms the baseline† PoseNet [26]. With one-shot refinement, our RAP_{ref} further reduces outdoor localization errors to below 10 cm.

Aachen Day-Night [53]. Benefiting from appearance augmentation, our RAP significantly reduces the localization error from 134 unit / 75.99° to 130 unit / 13.70°, outperforming other APR [10, 26] and SCR [6, 66] baselines, as shown in Table 3. This demonstrates the effectiveness of appearance diversity in handling extreme lighting changes.

7-Scenes [59]. As shown in Table 4, our RAP reduces translation error by 50% (10 cm → 5 cm) and rotation error by 41.36% (3.24° → 1.90°) on average compared to previous state-of-the-art single-frame APR methods. The only exception is *Heads*, where the rotation error is suboptimal. This scene consists of just two sequences—one for training and one for testing—potentially limiting the effectiveness of our augmentation in capturing scene variability. Meanwhile, RAP_{ref} further reduces localization error below 1 cm with one-shot refinement using our 3DGS. It also surpasses ACE [6] and its post-refinement variant, ACE + GSCPR [36]. Qualitative examples are shown in Fig. 5.

4.3. Ablation Study

We conduct ablation studies on the validation set of *Shop* in the Cambridge Landmarks dataset to investigate the impact of all the components in our RAP. Setup I, our baseline, consists of the same components as in PoseNet [26] and has been retrained for our experiments. In Setup II, we replace the feature extraction from VGG16 [60] to EfficientB0 [38], which enhances performance due to its superior feature representation, while they both exhibit poor performance due to the lack of data synthesis. In Setup III and IV, we explore the effectiveness of the designed pose augmentation and appearance augmentation, which bring notable improvements: translation error reduces from 103 cm to 75 cm, and rotation error from 3.52° to 3.14°. In Setup V and VI, we add regular convolutional layers and Pose Transformer between feature extraction and pose regression. Both improve performance due to the increasing parameters, but the Transformer achieves superior results by effectively handling long-term dependencies through attention mechanisms. Finally, in Setup VII, our adversarial discriminator effectively reduces the syn-to-real domain gap, allowing the model to learn better pose regression features from synthetic data and further reduce localization error.

†DFNet [10] results are omitted as we were unable to successfully train its NeRF component, likely due to the need for manual scene scaling within $[-\pi, \pi]$, which is tedious for diverse outdoor scenes.

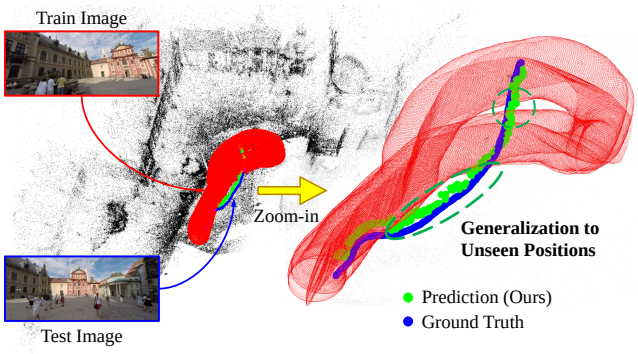

Figure 6. **Visualization of the training set distribution and results on *St. George's Basilica Building* [54].** The red hollow spheres, centered on the real images in the training set, indicate the potential locations of all synthetic images during training.

4.4. Discussion on Data Synthesis

Emerging Generalizability. Previously, APR has been understood to implicitly learn image retrieval [54], lacking the ability to successfully interpolate between training samples and generalize beyond them. To investigate how APR training is affected by increasing synthetic data, we trained RAP on the *St. George's Basilica Building* [54] and visualized the results in Fig. 6. Here, the translation perturbation was set to $\delta t = 350$ cm and the rotation perturbation to $\delta r = 60°$. Notably, the test set contained two regions entirely uncovered by the training set. Despite this, the model still closely predicts the test camera poses, demonstrating generalization ability beyond the original training positions.

We also learn from our experiments that reducing the rotation perturbation, such that the overlap between test and training views remains minimal, leads to high localization error. This is because the translation and rotation parameter space is inherently a $\text{SE}(3)$ manifold. Even if the translation remains fixed, significant rotation changes result in entirely different visual content in the images, naturally preventing the model from estimating poses of such unseen views, which correspond to a large distance on the $\text{SE}(3)$ manifold. Therefore, enabling generalization across a broader range of space is an important direction for future work.

Analyzing Generalization Boundaries. To evaluate the model's generalizability, we designed an experiment introducing a "void zone" centered on the test camera, where all real and synthetic data within this zone were excluded. The void zone was progressively expanded to determine the critical threshold at which the localization performance declines most significantly. Specifically, for *Shop*, we used 100% of the training set to ensure complete scene cov-

Table 6. **Exploring the generalization boundaries of the model with synthetic data.** Green, blue, and red percentages indicate the relative change in localization error (**Med Err**) compared to the scenario without a void zone.

	w/o Void Zone	w/ Void Zone (cm/°)				
		10/0.5	**20/1**	**30/1.5**	**50/2**	**80/2.5**
Med Err ↓	33/1.26	30/1.34	32/1.32	40/1.84	39/2.07	49/2.20
(rel. change)	0%/0%	-9%/6.3%	-3%/4.7%	21%/46.0%	18%/64.3%	48%/74.6%
Avg Err ↓	41/1.51	38/1.75	41/1.63	51/2.40	48/2.40	61/2.84
Max Err ↓	155/4.52	147/6.56	219/6.01	192/8.38	246/9.80	242/13.12
Min Err ↓	3/0.17	8/0.20	4/0.22	7/0.30	4/0.16	8/0.60

Table 7. **Ablation on different pose augmentation policies**.

Methods	*College*	*Hospital*	*Shop*	*Church*	Average
RAP (LENS [45])	73/1.15	126/1.87	71/3.37	128/3.50	100/22.47
RAP (Ours)	**52/0.90**	**87/1.21**	**33/1.48**	**53/1.52**	**56/1.28**

erage, with void zone ranges set as [10/0.5, 20/1, 30/1.5, 50/2, 80/2.5, 100/3] (cm/°). The results in Table 6 demonstrate a stepwise decline in performance. Initially, expanding the void zone has minimal impact on localization accuracy. However, at 30 cm / 1.5°, a sharp decrease in performance marks the model's generalization boundary.

Pose Augmentation Policy. We conduct experiments using a modified version of RAP, with the pose augmentation approach identical to that of LENS [45], as shown in the Table 7, where our method obtains superior performance. This may be because, although LENS's pose augmentation policy covers a broader spatial area than the training set, its synthetic data may have lower NVS image quality in many unseen regions, which could negatively affect APR training.

Density of Training Data. As shown in Table 8, our method with the proposed augmentation significantly reduces errors as the density of real training data increases from 20% to 80%. However, the localization accuracy remains almost unchanged from 80% to 100%, as the scene is already sufficiently covered. Notably, using 100% of the training data without augmentation can result in a significantly higher maximum error in translation, nearly double that with only 20% of the training data with augmentation, despite its limited spatial coverage. This suggests that our augmentation method successfully prevents overfitting to the training data, improving generalization to the test set.

Quality of Training Data. We evaluate the impact of synthetic image quality on model performance in Table 9, using 20% and 50% of the real data for pose regression. For *Shop*, it is evident that fewer training samples in 3DGS result in lower-quality rendered views (as indicated by lower PSNR), leading to suboptimal localization performance, particularly for rotation. Surprisingly, localization performance using only 20% of the data for training suboptimal 3DGS and pose regression surpasses the results obtained with 100% of the data without augmentation, as shown in Table 8. This experiment confirms the need for a robust NVS model and the proposed augmentation method in APR training.

Table 8. **Impact of the density of real training data.** Our augmentation improves the model's ability to generalize across the entire scene, although this effect has an upper limit.

Training Pose %	w/ Appearance & Pose Aug. (cm/°)					w/o Aug. (cm/°)	
	100%	**80%**	**60%**	**40%**	**20%**	**100%**	**50%**
Med Err ↓	33/**1.26**	**32**/1.27	37/1.90	57/2.23	87/3.65	98/3.75	104/4.17
Avg Err ↓	41/1.51	**40/1.50**	48/2.17	62/2.81	91/4.65	128/4.49	139/5.33
Max Err ↓	**155**/4.52	158/**4.09**	193/9.39	230/11.06	231/15.45	490/20.73	500/21.02
Min Err ↓	**3/0.17**	7/0.20	7/0.18	6/0.38	12/0.46	13/0.63	9/0.48

Table 9. **Impact of synthetic image quality.** Training with higher-quality synthetic images from advanced NVS models enhances localization performance.

3DGS Performance			Localization Performance (cm/°)				
% Images (Train)	**PSNR ↑ (Train)**	**PSNR ↑ (Test)**	**% Images (Train)**	**Med Err ↓**	**Avg Err ↓**	**Max Err ↓**	**Min Err ↓**
20%	29.08	15.98	**20%**	58/3.59	68/4.31	211/12.19	14/0.51
20%	29.08	15.98	**50%**	43/2.47	55/3.26	196/21.11	7/0.40
50%	26.88	17.55	**50%**	37/1.88	48/2.37	184/10.17	9/0.52
100%	24.60	18.30	**50%**	35/1.64	41/2.12	130/11.07	4/0.38

Table 10. **Ablation on different 3D representations.**

Methods	*College*	*Hospital*	*Shop*	*Church*	Average
DFNet [10]	73/2.37	200/2.98	67/2.21	137/4.03	119/2.90
DFNet_{GS}	102/2.31	137/8.08	77/3.92	123/4.68	110/4.75
RAP (Ours)	**52/0.90**	**87/1.21**	**33/1.48**	**53/1.52**	**56/1.28**

Different 3D Representations. We evaluate the impact of different 3D representations on the performance in Table 10. Trivially replacing NeRF with 3DGS in existing frameworks degrades performance due to 3DGS's inferior 3D consistency. This shows the effectiveness of our proposed joint training paradigm in RAP, which better utilizes diverse synthetic data to learn appearance-invariant features, rather than naively transferring from NeRF to 3DGS.

5. Conclusion

Summary. We address absolute pose regression with a robust two-branch joint training framework based on Transformer, coupled with an efficient data synthesis pipeline leveraging 3D Gaussian Splats (3DGS) to synthesize numerous posed images with diverse appearances as additional supervision. Our RAP achieves state-of-the-art localization performance, even under challenging appearance variations. Moreover, we thoroughly investigate the impact of synthesizing diverse data and present a novel perspective on APR: generalizability can emerge if the learning gap in APR is effectively addressed together with diverse data. We believe our RAP could be a promising starting point, and the experiments presented in the paper can provide useful insights for future research in this field.

Acknowledgment. This work was supported in part through NSF grants 2238968, 2121391, and 2024882, and the NYU IT High Performance Computing resources, services, and staff expertise. Yiming Li is supported by NVIDIA Graduate Fellowship (2024-2025).

Adversarial Exploitation of Data Diversity Improves Visual Localization

Supplementary Material

A. Pipeline Workflow

- **Stage 1: Appearance-Varying 3DGS**
  - **Input:** Sequences of RGB images $\boldsymbol{I}$ and corresponding camera poses $\boldsymbol{P}$.
  - **Output:** 3D appearance-varying Gaussians with deblurring ability.
  - **Loss:** $\mathcal{L}_1$, $\mathcal{L}_{\text{D-SSIM}}$, $\mathcal{L}_{\text{LPIPS}}$, $\mathcal{L}_{\boldsymbol{\mathcal{S}}}$, and optionally, $\mathcal{L}_{\text{depth}}$.
- **Stage 2: Two-Branch Joint APR Training**
  - **Branch-1**
    - **Input:** Real image-pose pair $(\boldsymbol{I}, \boldsymbol{P})$ and the corresponding synthesized image with the same pose $(\boldsymbol{I'}, \boldsymbol{P})$.
    - **Output:** Adjusted translation features $(\text{Adj}(\mathcal{F}_t(\boldsymbol{I}), \text{Adj}'(\mathcal{F}_t(\boldsymbol{I'})))$, adjusted rotation features $(\text{Adj}(\mathcal{F}_r(\boldsymbol{I})), \text{Adj}'(\mathcal{F}_r(\boldsymbol{I'})))$, and predicted poses $(\widehat{\boldsymbol{P}}, \widehat{\boldsymbol{P}'})$
    - **Loss:** Pose loss $\mathcal{L}^1_{\text{pose}}$, generator loss $\mathcal{L}_{\text{Gen}}$, and adversarial loss $\mathcal{L}_{\text{Dis}}$.
  - **Branch-2**
    - **Input:** Synthesized images with randomly blended appearances and perturbed poses $(\boldsymbol{I}_{\text{syn}}, \boldsymbol{P}_{\text{syn}})$.
    - **Output:** Predicted poses $\widehat{P_{\text{syn}}}$.
    - **Loss:** Pose loss $\mathcal{L}^2_{\text{pose}}$.
- **Stage-3: Post-Refinement**
  - **Input:** Rendered image from 3DGS using the query image and the initial pose estimated by the trained pose regressor.
  - **Output:** Final refined pose.
  - **Loss:** RANSAC-PnP [18, 19] solver on pixel-level matching between the rendered and query images.

B. 3D Gaussian Splatting Preliminary

Gaussian Splatting [27] is a promising approach for real-time novel view synthesis. By representing scenes as a set of 3D Gaussians, it retains the differentiable properties of volumetric radiance fields while offering more efficient optimization and higher-quality rendering compared to NeRF. The scene is defined through parameters such as position $\boldsymbol{\mu} \in \mathbb{R}^{K\times 3}$, covariance decomposed as rotation $\boldsymbol{q} \in \mathbb{R}^{K\times 4}$ and scaling $\boldsymbol{s} \in \mathbb{R}^{K\times 3}$, anisotropic color $\boldsymbol{c} \in \mathbb{R}^{K\times 3}$ modeled by sphere harmonics $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{K\times 16\times 3}$, and opacity $\boldsymbol{\alpha} \in \mathbb{R}^{K}$. During optimization, the scene representation is optimized by iteratively adjusting parameters through stochastic gradient descent, enabled by a differentiable rasterizer. This process is combined with adaptive density control to dynamically add or remove Gaussians based on the gradient of screen-space points corresponding to the Gaussians and opacity reset to reduce overfitting caused by floaters. The rendering process involves projecting 3D Gaussians onto the 2D image plane, sorting them by depth, and then applying α-blending to generate the final image. The render equation is:

$$\boldsymbol{C} = \sum_{i=1}^{N} T_i \alpha_i \boldsymbol{c}_i, \quad T_i = \prod_{j=1}^{i-1}(1-\alpha_j), \tag{13}$$

where $\boldsymbol{C} \in \mathbb{R}^3$ is each pixel's color and T_i is the transmittance. This approach significantly speeds up optimizing and rendering while achieving state-of-the-art visual quality.

C. MASt3R Preliminary

This section further elaborates on the background knowledge of MASt3R [29] mentioned in the paper. MASt3R grounds image matching tasks in 3D space to improve robustness and accuracy in challenging scenarios. Building on the DUSt3R [67] framework, MASt3R incorporates a new feature-matching head and a fast reciprocal matching algorithm, significantly enhancing performance for dense correspondences and camera pose estimation. It addresses the limitations of traditional 2D-based methods by leveraging dense 3D pointmaps and a coarse-to-fine matching strategy. Extensive evaluations demonstrate substantial gains in accuracy, computational efficiency, and generalizability, making MASt3R a robust solution for visual localization tasks.

D. Architecture Details

This section provides additional details regarding the network structure of RAP.

D.1. Feature Extraction

Our RAP pipeline first downsamples the input real images, adjusting the shorter side to approximately 240–360 px to enhance computational efficiency without losing much information. The downsampled images are then normalized and passed through the backbone network. For feature extraction, we utilize EfficientNet-B0 [38] as the backbone for multi-scale feature extraction. Translation feature $\mathcal{F}_t$ and rotation feature $\mathcal{F}_r$ are extracted from the third (`reduction_3`) and fourth (`reduction_4`) layers, with the number of feature channels being $C_t = 40$ and $C_r = 112$, which then are projected via 1×1 convolutions to align with the input channel dimension $D = 128$ of the proposed *Pose Transformer*.

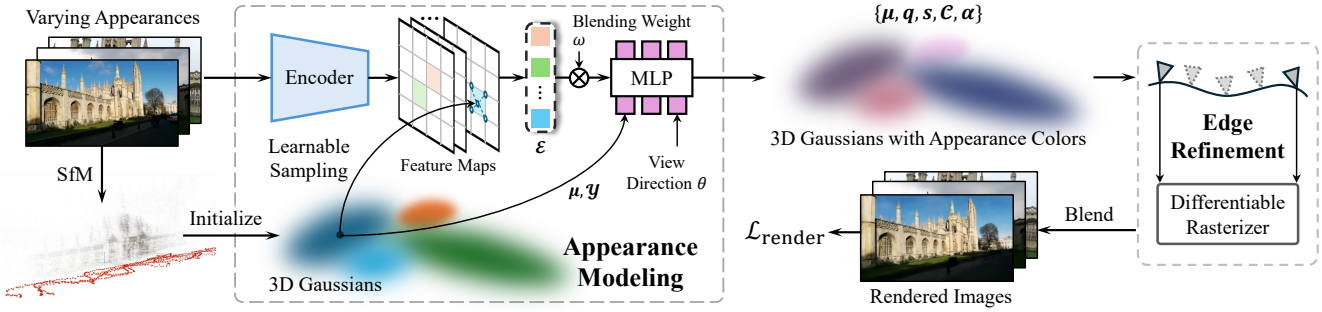

Figure I. **Overall illustration of appearance-varying 3DGS.** The framework models varying appearances using 3D Gaussians enhanced with appearance colors. It initializes 3D Gaussians from SfM data, refines their appearance by learnable sampling and blending weights computed via an encoder and MLP, and renders images by a differentiable rasterizer with edge refinement to minimize the rendering loss.

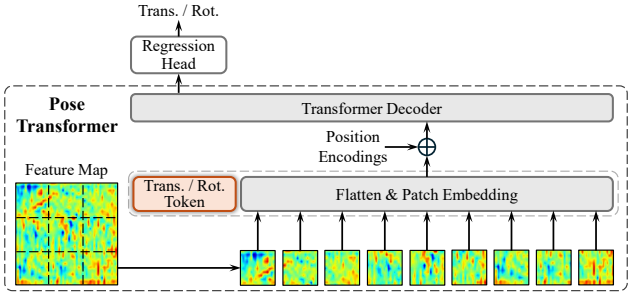

Figure II. **Structure of the Pose Transformer.** The feature map from the backbone is used as input, where we replace the `CLS` token in Vision Transformers (ViT) [16] with translation token `Trans` or rotation token `Rot` for the following regression head.

D.2. Pose Transformer

Relying on fine-grained local features, as done in previous works [10, 32], can hinder invariant feature learning due to image noise caused by dynamic objects and illumination changes. To overcome this, we leverage Transformer's robust ability to capture long-range dependencies, as illustrated in Fig. II. Taking the Cambridge Landmarks dataset [26] as an example, the original image resolution is 854×480. After downsampling and feature extraction, the resulting *translation feature map* (identical for the rotation feature map) has a shape of $[B, 112, H_t, W_t]$, where $H_t = 15$ and $W_t = 27$. A 1×1 convolutional layer is then applied to adjust the number of feature channels to 128, aligning it with the input dimension of the Transformer architecture. The *translation token* is a learnable parameter with 128 dimensions. The translation feature map is flattened and concatenated with the translation token, forming a matrix of size $[B, 128, H_t \times W_t + 1]$. This matrix, combined with positional encodings, is fed into the Transformer decoder, which consists of six layers, each containing multi-head self-attention with eight heads. Finally, the dimension corresponding to the translation token is extracted and passed to the regression head to predict the translation. For clarity, we describe the process using translation as an example, but the same approach is applied to rotation.

D.3. Adversarial Discriminator

We address the domain gap between synthetic and real images at the feature level by employing an adversarial discriminator. Specifically, the translation features $\mathcal{F}_t(\boldsymbol{I})$ and rotation features $\mathcal{F}_r(\boldsymbol{I})$ are first processed through the adjustment layers composed of two Conv-ReLU-BN layers, respectively, as $\text{Adj}(\mathcal{F}_t(\boldsymbol{I}))$ and $\text{Adj}'(\mathcal{F}_t(\boldsymbol{I'}))$, which align the channel dimensions to a consistent size of 128. The discriminator, implemented as a sequence of four convolutional layers with LeakyReLU activations and dropout, progressively reduces the spatial dimensions. The output is flattened and passed through a fully connected layer tailored for different datasets. Meanwhile, the MSE loss is applied in the discriminator, bridging the feature-level domain gap by effective adversarial learning.

D.4. Regression Head

The output features from the Transformer are fed into dedicated regression heads. For translation, the regressor outputs a 3-dimensional vector representing $[x, y, z]$ coordinates. For rotation, the regressor outputs a 6-dimensional vector, which is a continuous representation of a rotation matrix [74], which is subsequently converted into a 3×3 rotation matrix with Gram-Schmidt orthogonalization [3].

E. Implementation Details

E.1. Appearance-Varying 3DGS

The detailed pipeline of our Appearance-varying 3DGS is demonstrated in Fig. I. For challenging datasets with moving objects and camera motion blur, we extend the optimization iterations to 90,000 and adjust densification parameters and pruning behaviors according to the scene's size and complexity. For 7-Scenes [59], we use the pro-

Table I. **Metadata showing the number of images in the training and test sets for each scene.** The number of image sequences in each scene is indicated in parentheses. Different appearances across image sequences collected at different times pose challenges to modeling the environment and performing visual localization. More visualization can be found in Fig. IV, Fig. IX, and Fig. XII.

7-Scenes [59]	**Scenes**	***Chess*** (6)	***Fire*** (4)	***Heads*** (2)	***Office*** (10)	***Pumpkin*** (8)	***Kitchen*** (14)	***Stairs*** (6)	**Total**
	Train	4000	2000	1000	6000	4000	7000	2000	26000
	Test	2000	2000	1000	4000	2000	5000	1000	17000
Cambridge [26]	**Scenes**	***College*** (8)	***Hospital*** (9)	***Church*** (14)	***Shop*** (3)	-	-	-	**Total**
	Train	1220	895	1487	231	-	-	-	3833
	Test	343	182	530	103	-	-	-	1158
MARS [30]	**Scenes**	***"0011"*** (9)	***"0015"*** (5)	***"0037"*** (5)	***"0041"*** (5)	-	-	-	**Total**
	Train	792	788	771	819	-	-	-	3170
	Test	186	172	225	204	-	-	-	787

vided depth information to regularize 3DGS. We sample scenes that need edge refinement twice when optimizing each frame. The loss $\mathcal{L}$ is implemented as:

$$\mathcal{L} = \gamma_1\mathcal{L}_1 + \gamma_2\mathcal{L}_{\text{D-SSIM}} + \gamma_3\mathcal{L}_{\text{LPIPS}} + \gamma_4\mathcal{L}_{\mathcal{S}} + \gamma_5\mathcal{L}_{\text{depth}}, \quad (14)$$

where $\gamma_1 = 0.8$, $\gamma_2 = 0.2$, $\gamma_3 = 0.005$, $\gamma_4 = 0.001$, and γ_5 is decayed from 1 to 0.01 if depth regularization is enabled; otherwise, $\gamma_5 = 0$. $\boldsymbol{\mathcal{S}}$ is the learnable sampler mentioned in the paper. $\mathcal{L}_{\boldsymbol{\mathcal{S}}}$ is computed as:

$$\mathcal{L}_{\boldsymbol{\mathcal{S}}} = \frac{1}{n}\sum \text{ReLU}(|\boldsymbol{\mathcal{S}}| - 1). \quad (15)$$

E.2. Two-Branch Joint APR Training

The RAP is trained and tested with a batch size of 8–12 on NVIDIA RTX A6000 GPUs. To optimize training and save time, we employ an early stopping mechanism with a patience value of 200, and enable FP16 auto-mixed precision (AMP). The learning rate is reduced by a factor of 0.95 whenever the validation loss plateaus, with this adjustment made every 50 epochs. The loss weights used during training are $\beta_1 = 1, \beta_2 = 1, \beta_3 = 0.7$. We also include an additional VICReg loss [1] with the same weight with β_1 to mitigate the domain gap between the synthetic and real data. For every $N = 20$ epoch, we randomly generate the same number of views as the training sample size using our appearance and pose augmentations. The model generally converges after approximately 1000 epochs.

For scenes where the camera pose is close to surrounding objects, to reduce interference from augmented poses moving inside 3D Gaussian Splats and rendering low-quality images, while also avoiding manual adjustment of augmentation intensity, we filter out generated images with a BRISQUE [43] quality score ≥ 50 during augmentation in these scenes. However, this is not always effective and often weakens the augmentation effect. Enabling the augmentation policy to become learnable might be an interesting direction for future work.

E.3. Matching-Based Post Refinement

We only use MASt3R's coarse mode to obtain 2D-2D matches between the rendered RGB image and the query image to save time. Then, we back-project the rendered depth map from 3DGS into 3D space. In the following RANSAC-PnP [18, 19], we set the projection error to be 2 pixels. All other settings follow the defaults provided in the MASt3R repository [29].

Figure III. **Effectiveness of deblurring.** The images in the second column, generated by 3DGS with deblurring ability, exhibit clearer and sharper edges than those produced without deblurring.

F. Scene Metadata and Evaluation Metrics

All the benchmark metadata are shown in Table I. We use a widely accepted metric to assess and compare the localization performance of various methods: the median error in translation and rotation, defined as a cm and $b°$, respectively. In the main manuscript, we also report the mean, maximum, and minimum errors to statistically compare the performance distribution across different methods.

G. Cambridge Landmarks [26]

G.1. Effectiveness of Deblurring

Visual localization benchmarks are typically collected from video sequences, where motion blur between adjacent

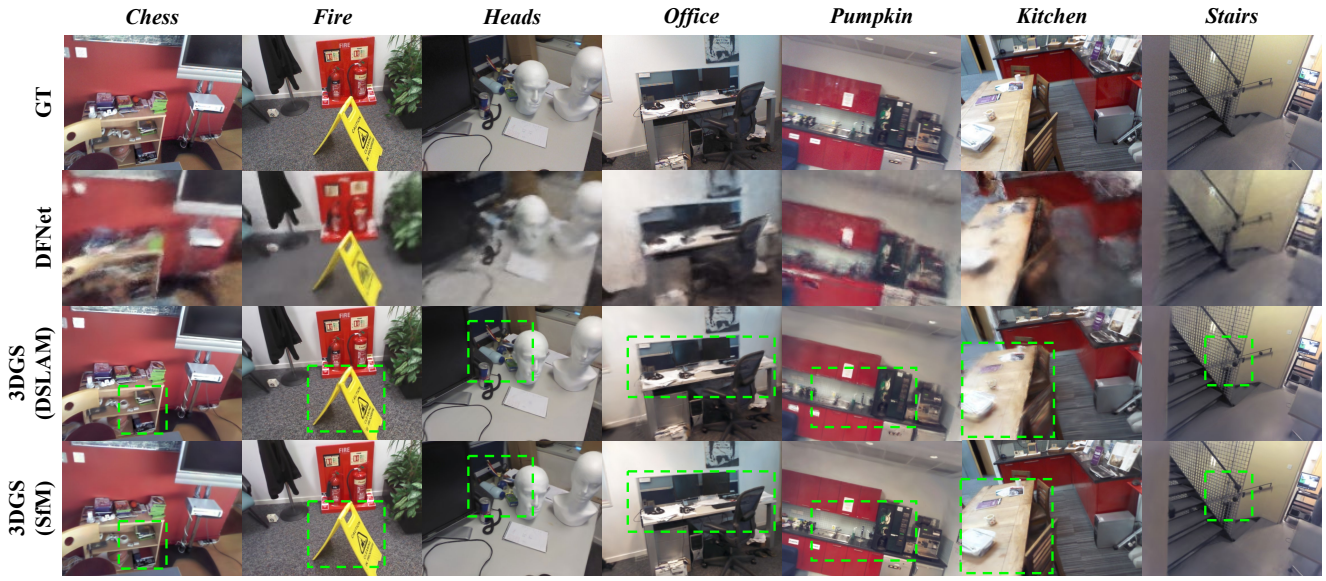

Figure IV. **Qualitative comparison of different NVS settings.** Our 3DGS achieves the highest visual quality with SfM ground truth poses, whereas DSLAM poses introduce noticeable blurriness, and the NeRF-based method delivers the worst results.

Table II. **Quantitative comparison of image quality between 3DGS using DSLAM [47] and SfM [55] poses.** SfM poses produce more realistic synthetic images with better consistency, as indicated by higher PSNR values that reflect higher image quality.

PSNR ↑	*Chess*	*Fire*	*Heads*	*Office*	*Pumpkin*	*Kitchen*	*Stairs*
DSLAM	19.98	19.04	17.23	20.89	19.20	18.92	18.93
SfM	**26.52**	**24.79**	**20.51**	**26.35**	**24.87**	**24.66**	**22.98**

frames is inevitable. This negatively affects both the optimization of 3DGS and localization performance. To address this, we incorporate a deblurring module when optimizing 3DGS to mitigate these effects. As shown in Fig. III, the deblurring module enhances modeling object edge details and removes artifacts, resulting in higher-quality data synthesis for APR training and post refinement.

G.2. 3DGS with Controllable Appearances

Figure IX presents images synthesized by our appearance-varying 3DGS on the Cambridge Landmarks dataset. The images rendered by our 3DGS exhibit finer details, such as sharper edges and textures, compared to those produced by DFNet [10], a NeRF-based method. These improvements contribute to better performance in both translation and rotation regression. Additionally, our 3DGS can model environments with varying lighting conditions using multiple image sequences, enabling seamless interpolation between them. This allows for synthesizing more diverse images, aiding RAP in learning robust and invariant features and further enhancing pose regression performance.

G.3. Additional Visualization of RAP_{ref}

The same process in RAP_{ref} on the Cambridge Landmarks dataset is shown in Fig. X, with results in Fig. XI. Compared to indoor scenes, localization errors in outdoor scenes are significantly larger. This is attributed to inherent limitations in scene scale and image resolution. Even when the visual matching between the real and synthesized images appears nearly perfect to humans, as indicated by the image continuity near the diagonal in Fig. XI, errors can still occur within a single pixel in the image coordinate system. For fairness, we use the same resolution as DFNet [10].

H. MARS [30]

H.1. Ground Truth Pose Details

The GPS/IMU poses provided in the dataset are inaccurate, so we use COLMAP [55] poses as the ground truth and compute the scaling factor relative to the GPS locations to calculate the metric translation error.

H.2. 3DGS with Controllable Appearances

The main challenges in driving scenarios include dynamic objects, such as vehicles and pedestrians, and dynamic environments with varying weather conditions. Fig. XII shows that our appearance-varying 3DGS successfully models variations in ambient lighting. Notably, it can also capture dynamic elements in the scene, such as vehicles on the road.

H.3. Additional Visualization of RAP_{ref}

Figure XIII and Fig. XIV illustrate the same post-refinement process and results of RAP_{ref} on the MARS dataset. Our

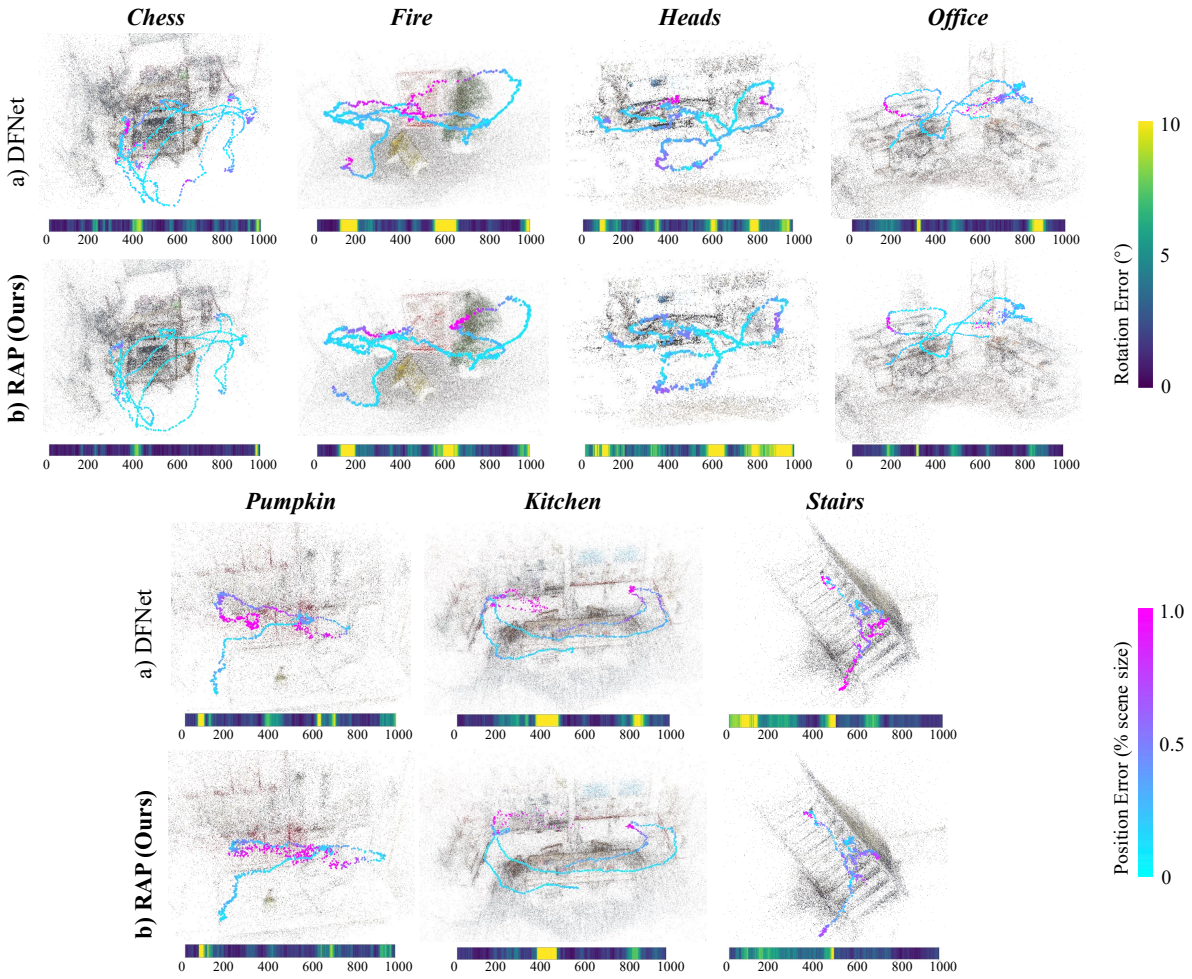

Figure V. **Visualization of estimated camera poses on the 7-Scenes dataset [59].** Translation and rotation errors are indicated by the color of the error bars. Our RAP framework more closely follows the ground truth trajectory with fewer outliers compared to DFNet [10]. The sequences visualized are: *Chess-seq-03*, *Fire-seq-04*, *Heads-seq-01*, *Office-seq-07*, *Pumpkin-seq-01*, *Kitchen-seq-14*, and *Stairs-all*.

model successfully handles the challenges of varying appearances in autonomous driving scenarios. As shown in Fig. XIV, although the ground truth images and rendered images along the diagonal often exhibit differences in appearance, this does not compromise localization accuracy, as evidenced by the continuity of the images.

I. Aachen Day-Night [53]

I.1. Subset Details

We used only a subset of the Aachen dataset due to compatibility issues with the pose ground truth. The full dataset contains images captured using various camera models with different resolutions, and its COLMAP ground truth assigns different camera intrinsics to each image. However, our APR network does not take the camera focal length as input. When the focal length is not consistent, it can lead to ambiguities. For example, close-up shots with a small focal length may appear similar to distant shots with a large focal length, despite different translations. Resolving this would require rerunning COLMAP with a unified camera model, which is computationally expensive for such a large scene, so we opted to use only a subset.

I.2. Visualization on Appearance Variation

Appearance variation in Aachen [53] is shown in Fig. VII.

J. 7-Scenes [59]

J.1. Visualization of Rendering Quality

Figure IV shows the image rendering results of our method compared to the DFNet [10] method across various scenes in the 7-Scenes dataset. DFNet consistently exhibits blurred edges and artifacts in all scenes, primarily due to the low resolution of voxel density sampling in NeRF. When the sampling points are insufficient, edge details become fuzzy.

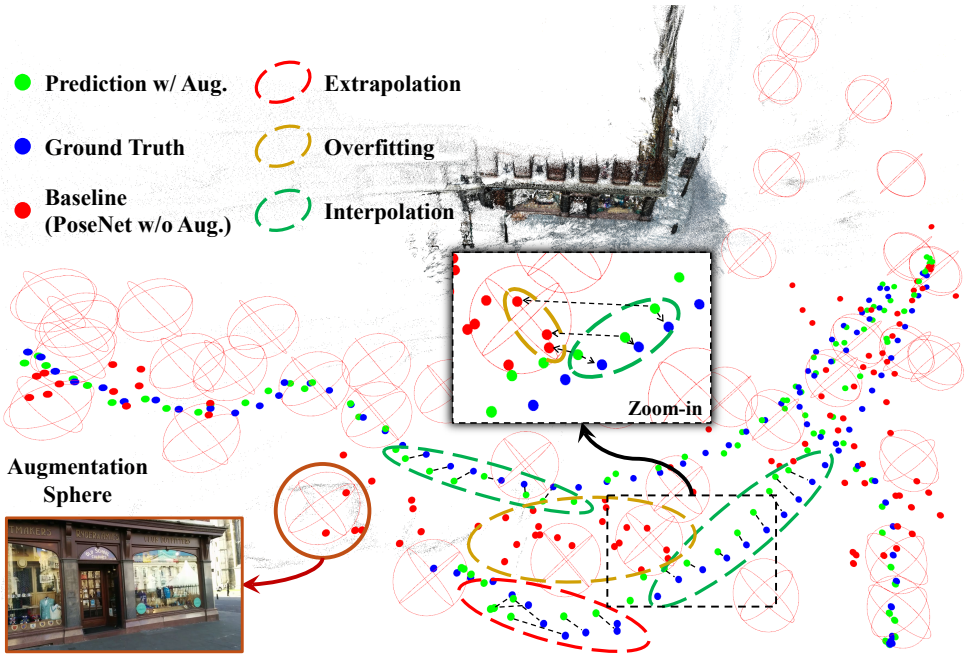

Figure VI. **Visualization of camera locations.** The red hollow spheres, centered on the real images in the training set, indicate the potential locations of all synthetic images during training. The blue dots, green dots, and red dots represent the ground truth, predictions by our RAP, and predictions by the baseline, respectively.

(a) $\omega = 0$ (b) $\omega = 0.5$ (c) $\omega = 1$ (d) $\omega = 1.5$ (e) $\omega = 2$

Figure VII. **Appearance variation in Aachen [53].**

In contrast, our method leverages the explicit 3DGS approach, successfully addressing this issue. The image quality achieved by our method is significantly better than that of NeRF-based methods. Furthermore, our deblurring technique ensures that object edges are clear and sharp, further enhancing the overall rendering quality.

J.2. Ground Truth Pose Details

In addition to evaluating performance using SfM ground truth poses of the 7-Scenes dataset, which enable synthesizing higher-quality images [11], we also provide results based on DSLAM [47] ground truth poses of the same dataset. As shown in Fig. IV, SfM poses yield more accurate results, while DSLAM poses introduce noticeable artifacts along object edges. The quantitative results in Table II compare the image quality metric (PSNR) for the two sets of poses, demonstrating a significant improvement in image quality with SfM poses. This enhancement further boosts the performance of APR. Furthermore, the table in the paper shows that RAP using SfM poses achieves lower localization errors, although RAP using DSLAM poses already delivers state-of-the-art performance.

J.3. Additional Visualization of RAP

We present qualitative comparisons on the subsets of the 7-Scenes dataset in Fig. V, comparing our RAP with DFNet [10]. In *Fire-seq-04*, *Pumpkin-seq-01*, and *Kitchen-seq-14*, our RAP avoids collapsing in certain regions, unlike DFNet, which generates a significant number of outliers. This demonstrates our RAP's strong generalizability.

J.4. Additional Visualization of RAP$_{\text{ref}}$

Figure XV illustrates the intermediate steps involved in post-refinement. Specifically, after obtaining the initial pose estimation of the query image from RAP, we render the corresponding image and depth map through 3DGS. Then, we use MASt3R [29] to calculate the pixel correspondences between the two images. As shown in Fig. XV, the matching lines before refinement are not sufficiently horizontal, indicating inaccuracies in the initial pose estimation. Next, we derive 2D-3D correspondences from the depth map and optimize the pose using a RANSAC-PnP [18, 19] solver. The final column of images demonstrates that the refined pose produces a rendered image almost indistinguishable from the original query image, with the matching lines now highly horizontal, demonstrating the improved accuracy of the pose estimation. As shown in Fig. XVI, the errors of our RAP$_{\text{ref}}$ are even less than 1 centimeter.

Table III. **Inference efficiency.** Measured on *Heads* with input images of size 320 × 240.

Method	PyTorch Mode	Avg FPS ↑
ACE [6]	With C++	50
RAP (Ours)	Eager	105
	Compiled	154
	Compiled `reduce-overhead`	187
	Compiled AMP	192
	Compiled `reduce-overhead` AMP	279

K. Emerging Generalization in APR

We experimented on *Shop* using only 20% of the real training set with our synthetic data, as shown in Fig. VI. We see that the training set with synthetic data, represented by the red hollow spheres, does not fully cover the test set spatially. Despite this, the model still closely predicts the test camera poses, demonstrating generalization ability beyond the original training positions.

L. Inference Efficiency

As shown in Table III, our Python prototype of RAP achieves approximately 279 FPS with the `reduce-overhead` mode of `torch.compile`[‡] and AMP enabled on a laptop equipped with an NVIDIA RTX 4060 GPU running at 30 W and an Intel Core i9-13900H CPU, demonstrating real-time inference performance on compact devices. For RAP_{ref}, the post-refinement time per frame is 0.5 s on the same device, including RAP inference, 3DGS rendering with `gsplat` [69], MASt3R matching, and RANSAC-PnP [18, 19] solving using OpenCV. During this process, the GPU power consumption can reach 70–80 W. Please see our code for additional implementation details.

M. Failure Cases

Fig. VIII presents several failure cases encountered during evaluation. Occlusions pose the most significant challenge for APR, particularly when dynamic objects are present. For example, in the second-row images, tree branches—absent in the 3DGS-synthesized image—appear during the inference stage, disrupting feature extraction. Additionally, textureless patterns in the image can degrade APR performance. For instance, in the third row, the stark contrast between the featureless sky and the building's underexposed color creates ambiguities, posing challenges for feature extraction, potentially misleading the regression head, and impacting localization accuracy.

[‡]Timing measured using the function provided in https://pytorch.org/tutorials/intermediate/torch_compile_tutorial.html; dataloader time is excluded.

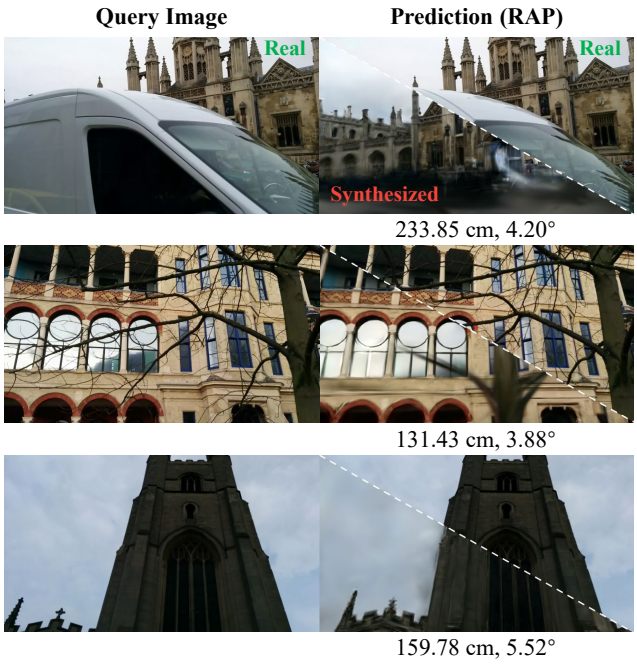

Figure VIII. **Failure cases.** The primary reason for localization failure is occlusion, as shown in the first two rows. Additionally, textureless regions in the query image, such as the sky, can also result in significant errors.

N. Limitations and Future Work

Like other APR approaches, our method has yet to surpass geometry-based techniques in accuracy, and per-scene training remains time-consuming. We also observe accuracy loss when training on the metric scale in large scenes. Additionally, current APR methods do not account for camera intrinsics and are sensitive to input image resolution, leading to accuracy degradation when the testing resolution differs from training. Future directions include efficiently training stronger APR models with geometric priors, leveraging temporal information, and integrating powerful vision foundation models [49]. Generalizing to dynamic environments with fewer training samples is also a promising research avenue.

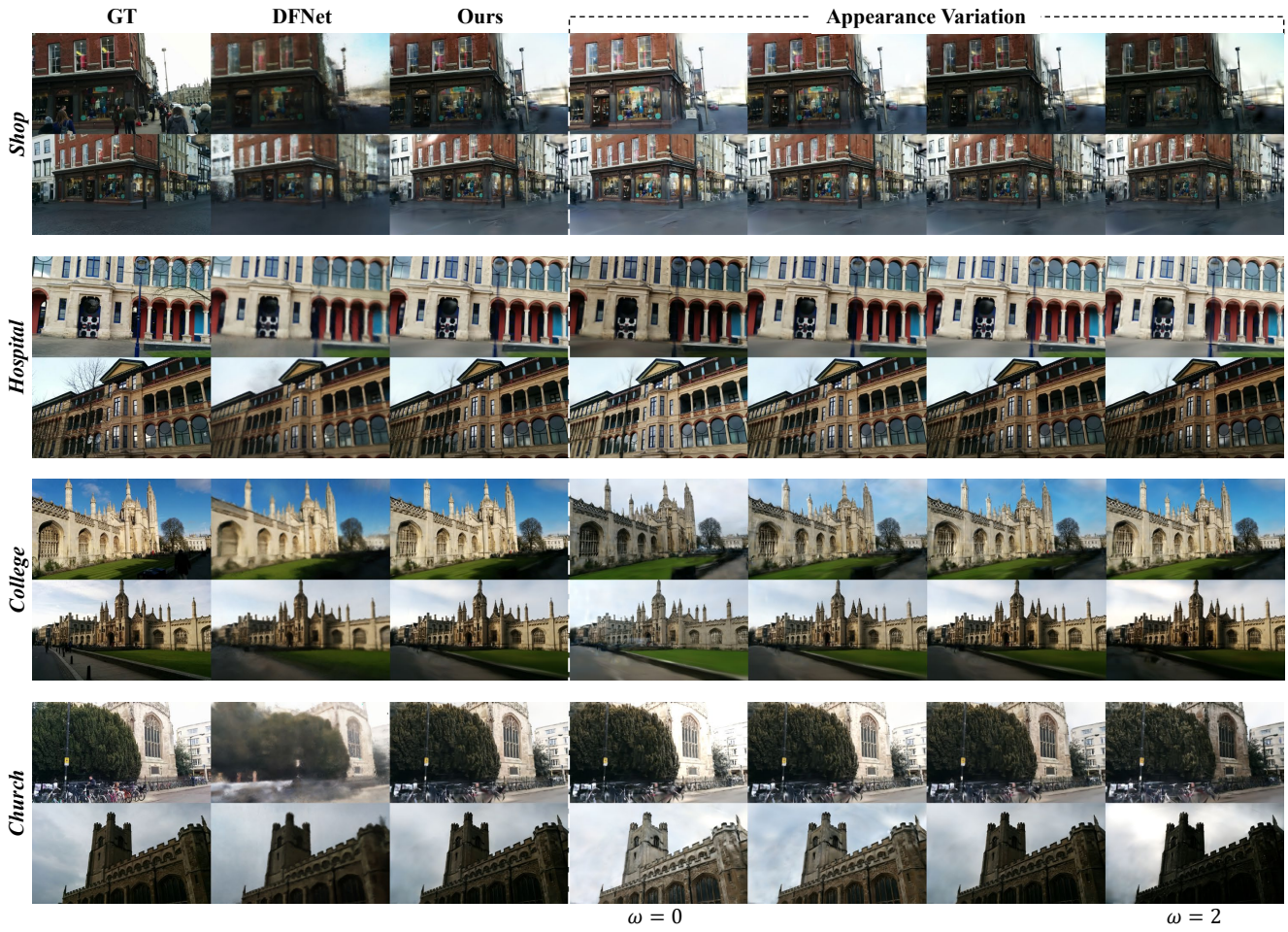

Figure IX. **Synthetic images with varying appearances on the Cambridge Landmarks dataset [26].** The appearances of synthetic images can be arbitrarily generated using different blending weights ω, ranging from 0 to 2.

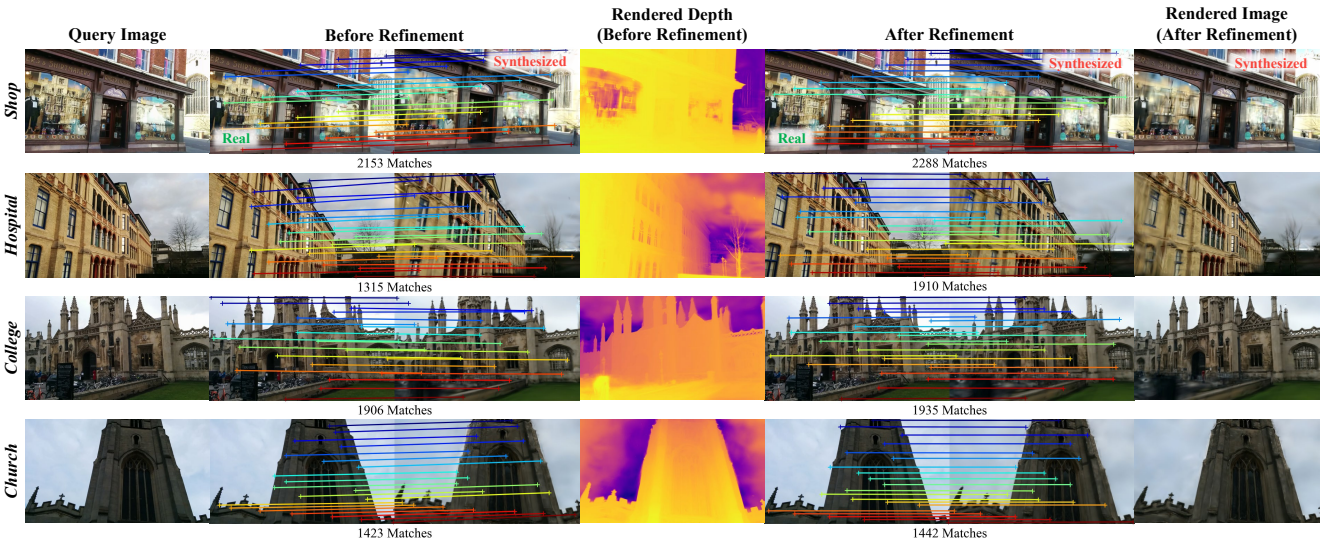

Figure X. **Visualization of the post-refinement pipeline on the Cambridge Landmarks dataset [26].** Starting with the query image, we first obtain its initial pose from RAP, render it using 3DGS, and generate matches. The lines before refinement are not sufficiently horizontal due to inaccuracies in the initial pose. Next, we back-project the rendered depth to 3D and use RANSAC-PnP [18, 19] to compute a refined pose, which is then tested by rendering and matching again. The matches after refinement are horizontal, indicating that the refined poses are more accurate.

Figure XI. **Visualization of localization errors on the Cambridge Landmarks dataset [26].** In each sub-figure, a diagonal boundary separates the "Predicted" (rendered from the refined pose) and "GT" (ground truth) sections. Smooth alignment along this boundary demonstrates RAP_{ref}'s improved pose accuracy.

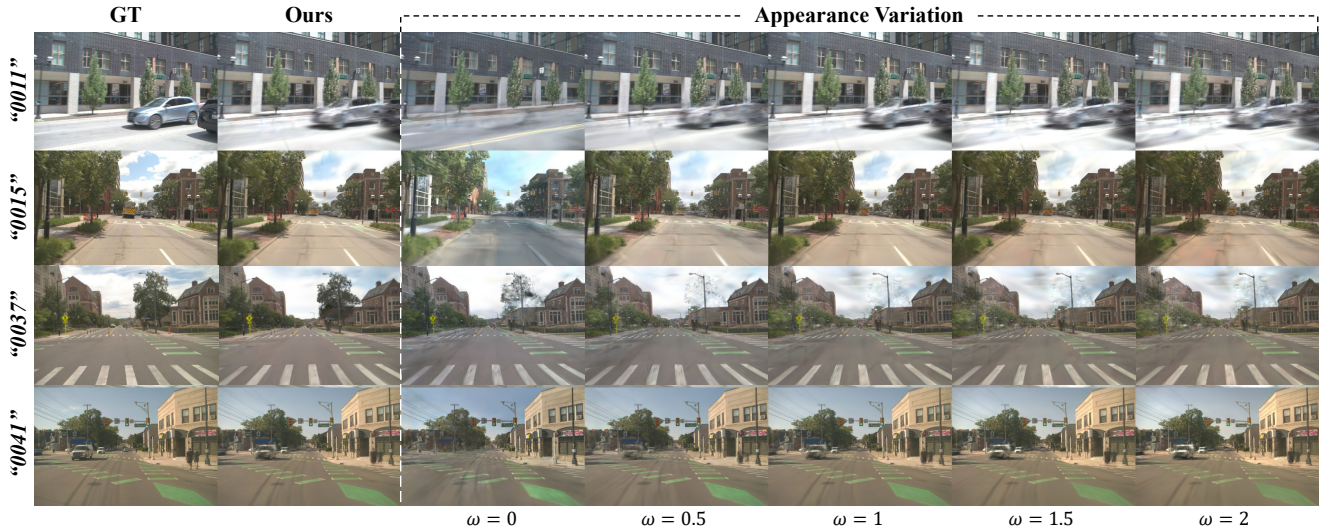

Figure XII. **Synthetic images with varying appearances on the MARS dataset [30].** The appearances of the synthetic images can be arbitrarily generated using different blending weights ω, ranging from 0 to 2.

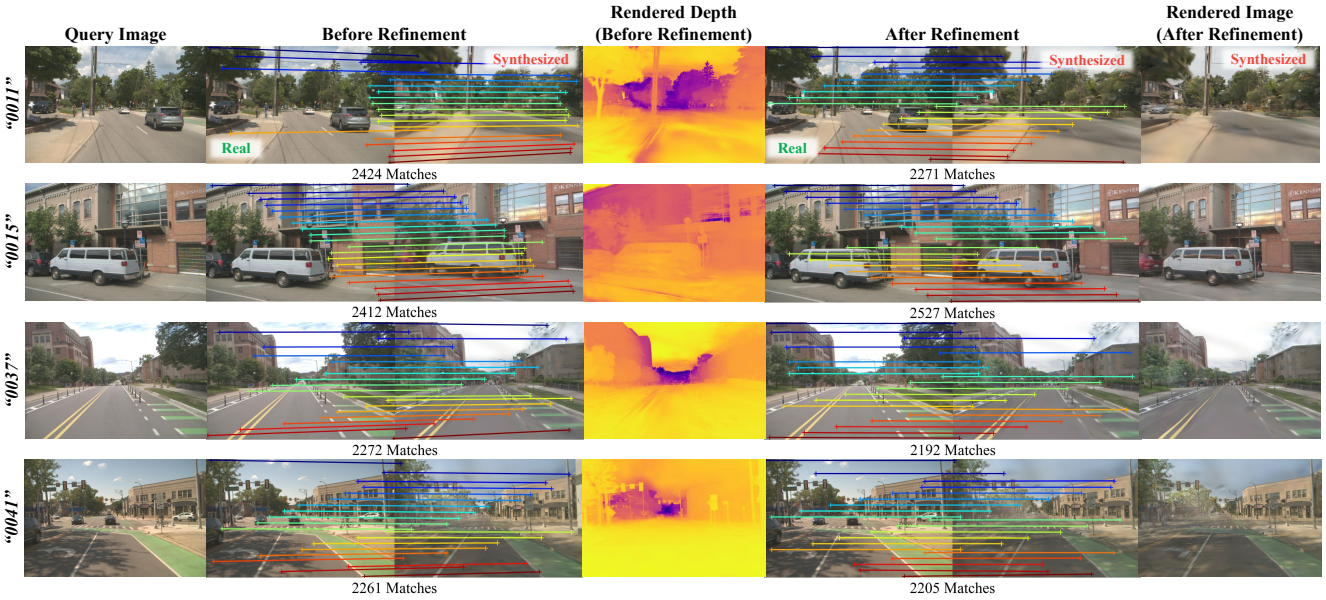

Figure XIII. **Visualization of the post-refinement pipeline on the MARS dataset [30].** Starting with the query image, we first obtain its initial pose from RAP, render it using 3DGS, and generate matches. The lines before refinement are not sufficiently horizontal due to inaccuracies in the initial pose. Next, we back-project the rendered depth to 3D and use RANSAC-PnP [18, 19] to compute a refined pose, which is then tested by rendering and matching again. The matches after refinement are horizontal, indicating that the refined poses are more accurate. Moreover, the rendered depth maps illustrate that our appearance-varying 3DGS successfully reconstructs the scene's geometric information, a critical factor in ensuring accurate 2D-3D correspondences.

Figure XIV. **Visualization of the localization errors on the MARS dataset [30].** In each sub-figure, a diagonal boundary separates the "Predicted" (rendered from the refined pose) and "GT" (ground truth) sections. Smooth alignment along this boundary demonstrates RAP_{ref}'s improved pose accuracy.

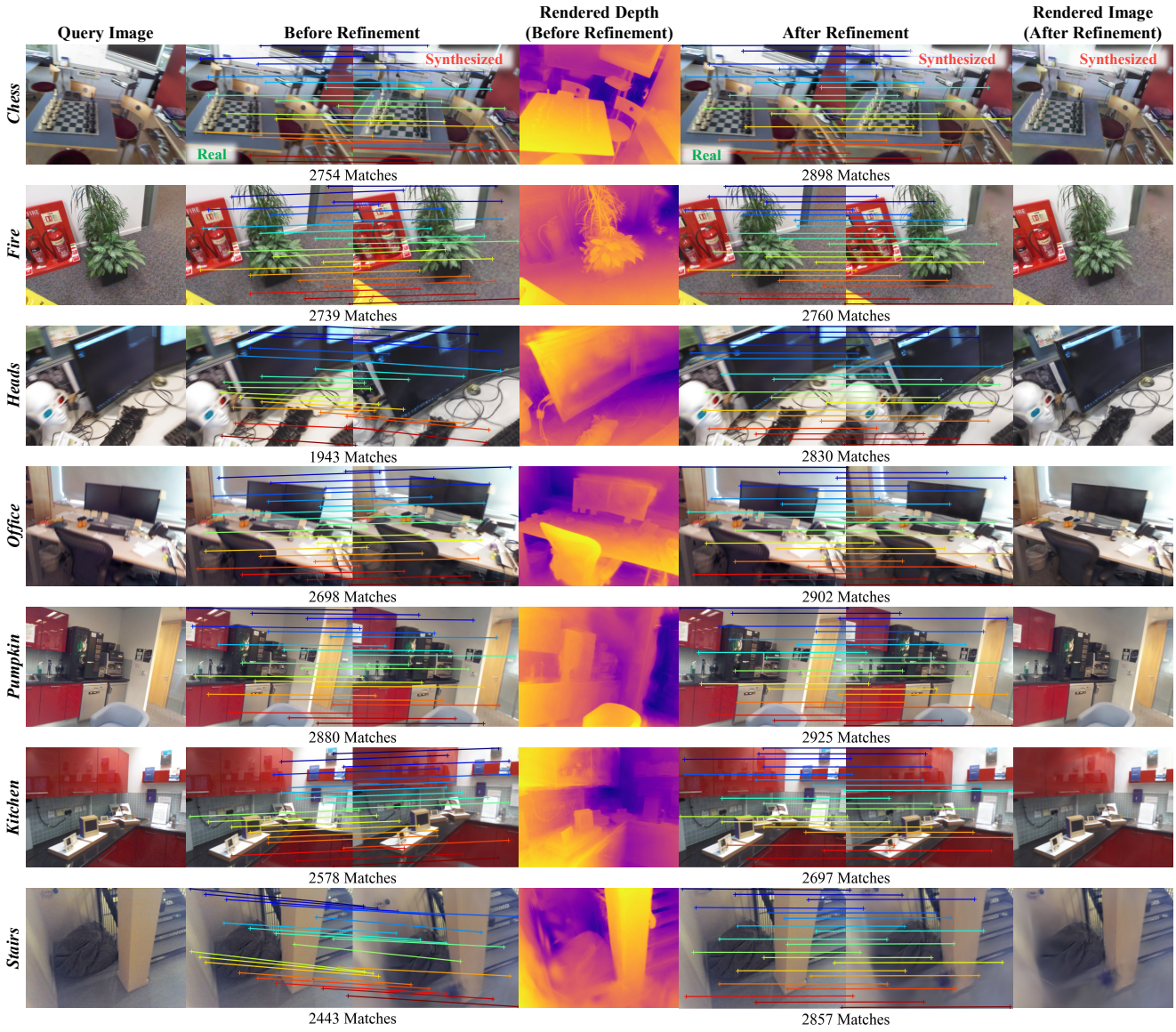

Figure XV. **Visualization of the post-refinement pipeline on the 7-Scenes dataset [59].** Starting with the query image, we first obtain its initial pose from RAP, render it using 3DGS, and generate matches. The lines before refinement are not sufficiently horizontal due to inaccuracies in the initial pose. Next, we back-project the rendered depth to 3D and use RANSAC-PnP [18, 19] to compute a refined pose, which is then tested by rendering and matching again. The matches after refinement are horizontal, indicating that the refined poses are more accurate. Moreover, the rendered depth maps illustrate that our appearance-varying 3DGS successfully reconstructs the scene's geometric information, a critical factor in ensuring accurate 2D-3D correspondences.

Figure XVI. **Visualization of localization errors on the 7-Scenes dataset [59].** In each sub-figure, a diagonal boundary separates the "Predicted" (rendered from the refined pose) and "GT" (ground truth) sections. Smooth alignment along this boundary demonstrates RAP_{ref}'s improved pose accuracy.

References

[1] Adrien Bardes, Jean Ponce, and Yann Lecun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022. 3

[2] Joydeep Biswas and Manuela Veloso. Depth camera based indoor mobile robot localization and navigation. In *2012 IEEE International Conference on Robotics and Automation*, pages 1697–1702, 2012. 1

[3] Åke Björck. Solving linear least squares problems by gram-schmidt orthogonalization. *BIT Numerical Mathematics*, 7 (1):1–21, 1967. 2

[4] 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. 2, 5, 6

[5] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6684–6692, 2017. 1, 5

[6] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5044–5053, 2023. 1, 2, 4, 5, 6, 7

[7] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2018. 1, 2, 5, 6

[8] David M. Chen, Georges Baatz, Kevin Koser, Sam S. Tsai, Ramakrishna Vedantham, Timo Pylvanainen, Kimmo Roimela, Xin Chen, Jeff Bach, Marc Pollefeys, Bernd Girod, and Radek Grzeszczuk. City-scale landmark identification on mobile devices. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2011. 2

[9] Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-posenet: Absolute pose regression with photometric consistency. In *2021 International Conference on 3D Vision (3DV)*, 2021. 2

[10] Shuai Chen, Xinghui Li, Zirui Wang, and Victor A Prisacariu. Dfnet: Enhance absolute pose regression with direct feature matching. In *European Conference on Computer Vision*, pages 1–17. Springer, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[11] Shuai Chen, Yash Bhalgat, Xinghui Li, Jia-Wang Bian, Kejie Li, Zirui Wang, and Victor Adrian Prisacariu. Neural refinement for absolute pose regression with feature synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20987–20996, 2024. 1, 4, 5, 6

[12] Shuai Chen, Tommaso Cavallari, Victor Adrian Prisacariu, and Eric Brachmann. Map-relative pose regression for visual re-localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20665–20674, 2024. 6

[13] Manuela Chessa, Chiara Bassano, and Fabio Solari. Detection and localization of changes in immersive virtual reality. In *International Conference on Image Analysis and Processing*, pages 121–132, 2023. 1

[14] Hiba Dahmani, Moussab Bennehar, Nathan Piasco, Luis Roldao, and Dzmitry Tsishkou. Swag: Splatting in the wild images with appearance-conditioned gaussians. In *European Conference on Computer Vision*, pages 325–340, 2025. 2

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF International Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 3, 2

[17] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8092–8101, 2019. 2

[18] 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. 2, 6, 1, 3, 7, 8, 10, 12

[19] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003. 2, 6, 1, 3, 7, 8, 10, 12

[20] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE/CVF International Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 1

[21] Hugo Germain, Daniel DeTone, Geoffrey Pascoe, Tanner Schmidt, David Novotny, Richard Newcombe, Chris Sweeney, Richard Szeliski, and Vasileios Balntas. Feature query networks: Neural surface description for camera pose refinement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5071–5081, 2022. 1, 5, 6

[22] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2

[23] Simon J. Julier and Jeffrey K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *Signal Processing, Sensor Fusion, and Target Recognition VI*, pages 182 – 193, 1997. 6

[24] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[25] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5974–5983, 2017. 4

[26] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *IEEE international conference on computer vision*, pages 2938–2946, 2015. 1, 2, 4, 5, 6, 7, 3, 8, 9

[27] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transaction on Graphics*, 42 (4), 2023. 1, 2, 3

[28] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 5, 6

[29] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. *arXiv preprint arXiv:2406.09756*, 2024. 6, 1, 3

[30] Yiming Li, Zhiheng Li, Nuo Chen, Moonjun Gong, Zonglin Lyu, Zehong Wang, Peili Jiang, and Chen Feng. Multiagent multitraversal multimodal self-driving: Open mars dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22041–22051, 2024. 4, 5, 6, 3, 10, 11

[31] Yiming Li, Zehong Wang, Yue Wang, Zhiding Yu, Zan Gojcic, Marco Pavone, Chen Feng, and Jose M. Alvarez. Memorize what matters: Emergent scene decomposition from multitraverse. In *Advances in Neural Information Processing Systems*, 2024. 2

[32] Jingyu Lin, Jiaqi Gu, Bojian Wu, Lubin Fan, Renjie Chen, Ligang Liu, and Jieping Ye. Learning neural volumetric pose features for camera localization. In *European Conference on Computer Vision*, 2024. 1, 2, 3, 4, 5, 6

[33] Jiaqi Lin, Zhihao Li, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Jiayue Liu, Yangdi Lu, Xiaofei Wu, Songcen Xu, Youliang Yan, and Wenming Yang. Vastgaussian: Vast 3d gaussians for large scene reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5166–5175, 2024. 2

[34] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *IEEE/CVF International Conference on Computer Vision*, pages 17627–17638, 2023. 2

[35] Changkun Liu, Shuai Chen, Yukun Zhao, Huajian Huang, Victor Prisacariu, and Tristan Braud. Hr-apr: Apr-agnostic framework with uncertainty estimation and hierarchical refinement for camera relocalisation. In *2024 IEEE International Conference on Robotics and Automation*, pages 8544–8550, 2024. 1, 5, 6

[36] Changkun Liu, Shuai Chen, Yash Sanjay Bhalgat, Siyan HU, Ming Cheng, Zirui Wang, Victor Adrian Prisacariu, and Tristan Braud. GS-CPR: Efficient camera pose refinement via 3d gaussian splatting. In *The Thirteenth International Conference on Learning Representations*, 2025. 5, 6, 7

[37] Liu Liu, Hongdong Li, and Yuchao Dai. Efficient global 2d-3d matching for camera localization in a large-scale 3d map. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 2

[38] Liu Liu, Hongdong Li, and Yuchao Dai. Efficient global 2d-3d matching for camera localization in a large-scale 3d map. In *IEEE International Conference on Computer Vision*, pages 2372–2381, 2017. 5, 7, 1

[39] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *IEEE/CVF International Conference on Computer Vision and Pattern Recognition*, pages 2794–2802, 2017. 4

[40] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[41] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1

[42] 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, 2020. 1

[43] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21 (12):4695–4708, 2012. 3

[44] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Coordinet: uncertainty-aware pose regressor for reliable vehicle localization. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2229–2238, 2022. 2, 5, 6

[45] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Lens: Localization enhanced by nerf synthesis. In *Conference on Robot Learning*, pages 1347–1356, 2022. 1, 2, 5, 6, 8

[46] Arthur Moreau, Nathan Piasco, Moussab Bennehar, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Crossfire: Camera relocalization on self-supervised features from an implicit representation. In *IEEE/CVF International Conference on Computer Vision*, pages 252–262, 2023. 1, 5, 6

[47] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136, 2011. 4, 6

[48] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *IEEE international conference on computer vision*, pages 3456–3465, 2017. 2

[49] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, pages 1–31, 2024. 7

[50] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *IEEE/CVF Conference*

on Computer Vision and Pattern Recognition, pages 12716–12725, 2019. 2

[51] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4938–4947, 2020.

[52] 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

[53] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8601–8610, 2018. 4, 5, 7, 6

[54] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3302–3312, 2019. 1, 2, 7

[55] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 4

[56] Yoli Shavit and Yosi Keller. Camera pose auto-encoders for improving pose regression. In *European Conference on Computer Vision*, pages 140–157, 2022. 5, 6

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

[58] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *IEEE/CVF International Conference on Computer Vision*, pages 2733–2742, 2021. 1, 5, 6

[59] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013. 4, 6, 7, 2, 3, 5, 12, 13

[60] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3, 7

[61] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM Transactions on Graphics*, pages 835–846, 2006. 2

[62] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7199–7209, 2018. 2

[63] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019. 3

[64] Gabriele Trivigno, Carlo Masone, Barbara Caputo, and Torsten Sattler. The unreasonable effectiveness of pre-trained features for camera pose refinement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12798, 2024. 1, 5, 6

[65] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. In *AAAI Conference on Artificial Intelligence*, pages 10393–10401, 2020. 6

[66] Fangjinhua Wang, Xudong Jiang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. Glace: Global local accelerated coordinate encoding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21562–21571, 2024. 1, 2, 5, 6, 7

[67] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 1

[68] Chen Wenbo and Liu Ligang. Deblur-gs: 3d gaussian splatting from camera motion blurred images. In *ACM in Computer Graphics and Interactive Techniques*, 2024. 2, 3

[69] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for Gaussian splatting. *arXiv preprint arXiv:2409.06765*, 2024. 7

[70] Dongbin Zhang, Chuming Wang, Weitao Wang, Peihao Li, Minghan Qin, and Haoqian Wang. Gaussian in the wild: 3d gaussian splatting for unconstrained image collections. In *European Conference on Computer Vision*, 2024. 2, 3

[71] Boming Zhao, Luwei Yang, Mao Mao, Hujun Bao, and Zhaopeng Cui. Pnerfloc: Visual localization with point-based neural radiance fields. In *AAAI Conference on Artificial Intelligence*, pages 7450–7459, 2024. 1, 5

[72] Enliang Zheng and Changchang Wu. Structure from motion using structure-less resection. In *IEEE International Conference on Computer Vision*, pages 2075–2083, 2015. 5

[73] Qunjie Zhou, Maxim Maximov, Or Litany, and Laura Leal-Taixé. The nerfect match: Exploring nerf features for visual localization. *arXiv preprint arXiv:2403.09577*, 2024. 1, 5

[74] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019. 2