Disambiguating 2D-3D Correspondences in Gaussian Splatting-based Feature Fields for Visual Localization

Miso Lee[1] **Sangeek Hyun**[2] **Yerim Jeon**[1] **Jae-Pil Heo**[1,†]
[1]Sungkyunkwan University [2]NAVER LABS

Abstract

While Gaussian Splatting-based Feature Fields (GSFFs) have shown promise for visual localization, this paper highlights that photometrically optimized GSFFs are inherently not suitable for 2D-3D matching. The volumetric extent of each Gaussian induces many-to-one pixel-to-point mappings that destabilize PnP-based pose estimation, while photometric optimization gives rise to superfluous Gaussians devoid of multi-view consistency. To address these issues, we propose **SplitGS-Loc**, a GSFFs construction framework for localization, which disambiguates 2D-3D correspondences by exploiting Gaussian attributes. Our key design, Mixture-of-Gaussians-based splitting, decomposes each Gaussian into smaller Gaussians, replacing ambiguous many-to-one with precise one-to-one correspondences. In parallel, we exploit composition weights from GS rasterization to select Gaussians that significantly and consistently contribute across multiple views and aggregate discriminative features through strong pixel-Gaussian associations. The resulting compact yet discriminative feature fields enable stable PnP convergence, achieving competitive performance on localization benchmarks. Extensive experiments validate that SplitGS-Loc extends the utility of photometric GSFFs to accurate and efficient localization by exploiting Gaussian attributes, without per-scene training for feature fields or iterative pose refinement. The code is available at: `https://github.com/misoleekr/SplitGS-Loc`.

1 Introduction

Visual localization is a fundamental computer vision task that estimates the camera pose from an image, providing the foundation for autonomous driving, robotics, and AR/VR systems. When a predefined 3D map is given, the camera pose can be obtained by matching 2D pixels with 3D points and using Perspective-n-Points (PnP) [1] with RANSAC [2]. Nowadays, as more expressive 3D representations like Gaussian Splatting (GS) [3] and 3D feature fields [4–6] have been actively explored, localization performance has been further advanced with GS-based Feature Fields (GSFFs) [7–9]. GSFFs are usually constructed by adding features to Gaussians with feature rendering loss [6–8], or jointly optimizing colors and features [5, 9]. This allows camera pose refinement through 2D-2D matching between query image features and rendered image features.

Recently, STDLoc [9] enables direct 2D-3D matching by sampling both query image features and GSFFs, reducing the computational overhead during inference. However, it demands hours of multi-stage optimization per scene; even after optimizing GSFFs, a scene-specific detector is required to align 2D keypoints with sampled Gaussians as GSFFs alone are not suitable for 2D-3D matching. Moreover, to achieve high localization accuracy, it further stores the full GSFFs and performs iterative dense feature-based pose refinement. Thus, although STDLoc establishes an effective sparse-to-dense localization pipeline, it still relies on additional query-side detection and dense 2D-2D refinement.

[†]Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

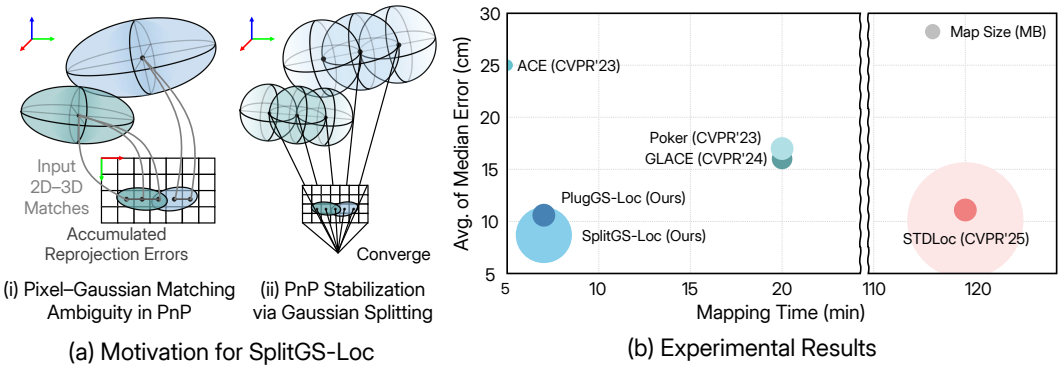


Figure 1: **Motivation and experimental results.** (a) Each 3D Gaussian covers a volumetric region and can thus correspond to multiple image pixels. However, PnP takes 2D-3D point correspondences as input; when each Gaussian is approximated by a single 3D point (e.g., its center), this leads to many-to-one correspondences. We resolve this by decomposing Gaussians into smaller ones, increasing valid one-to-one correspondences and stabilizing PnP-based localization. (b) The figure illustrates the trade-off between mapping efficiency and localization accuracy on Cambridge Landmarks [10]. While recent GSFFs-based methods achieve superior accuracy but sacrifice mapping efficiency, whereas our proposed frameworks attain a favorable balance by disambiguating pixel-Gaussian correspondences.

We discover that this inefficiency can be resolved by directly addressing the photometrically optimized Gaussians for 2D-3D matching. First, we reveal that the volumetric extent of Gaussians entails a structural mismatch with PnP, which requires strict point-to-point correspondences as input. As illustrated in Fig. 1 (a), when multiple 2D pixels correspond to the same Gaussian and are all mapped to a single 3D point, these many-to-one correspondences lead to geometric inconsistency. Consequently, PnP with RANSAC becomes unstable and ultimately outputs a suboptimal pose. In addition, numerous redundant Gaussians are spawned to capture view-dependent photometric effects [3, 11, 12]. Hence, they serve no meaningful role in 2D-3D matching under novel viewpoints due to the lack of multi-view consistency.

To this end, we propose a localization-specialized GSFFs construction framework, **SplitGS-Loc**, which disambiguates pixel-Gaussian correspondences by utilizing Gaussian attributes. Central to our approach is Mixture-of-Gaussians-based splitting, which decomposes each Gaussian along its major axis. This fast, deterministic, and feature-independent decomposition yields fine-grained Gaussians that serve as the foundation for stable 2D-3D matching. Yet, for those Gaussians to produce valid one-to-one matches at inference, each must be paired with its own distinctive feature.

This motivates us to devise a direct feature lifting pipeline, PlugGS-Loc, which exploits composition weights derived from the rasterization process. Specifically, given training views and photometric GS, it proceeds to identify salient Gaussians that significantly and consistently contribute to pixel rasterization across multiple views, and then assign to each selected Gaussian a pixel-level feature aggregated through strong pixel-Gaussian associations. This prevents feature blurring due to rendering-based optimization [13, 14] and promotes multi-view consistency.

Altogether, SplitGS-Loc disambiguates 2D-3D correspondences in GSFFs by alleviating many-to-one matching ambiguity through Gaussian decomposition and assigning a distinctive feature to each split Gaussian. As a result, we achieve competitive performance across various indoor and outdoor scenes [10, 15]. Extensive experiments demonstrate that reducing many-to-one pixel-Gaussian matches leads to stable PnP convergence, yielding faster runtime and better accuracy. From a broader perspective, this work reframes GSFFs from exhaustive 3D maps to deployable ones, restoring the balance between performance and practicality as illustrated in Fig. 1 (b).

2 Related Work

Classical methods follow a multi-stage pipeline [16–22]. Given a query image, (i) similar reference views are first retrieved from an image database [23, 24], (ii) local feature extraction and matching [25–27] are performed between the query and retrieved images, (iii) the resulting 2D-2D correspondences

are lifted to 2D-3D correspondences, as each reference image is registered to a 3D map via Structure-from-Motion (SfM), and (iv) the camera pose is estimated using the Perspective-n-Point (PnP) [1] algorithm within a RANSAC loop [2]. Although this pipeline provides strong performance, it relies on a large-scale image database and 3D map, together with multiple stages.

Deep learning-based methods have been developed in response to the need for efficiency in practical applications, aiming toward more streamlined pipelines. Scene Coordinate Regression (SCR) methods [28–31] train a network to predict a 3D scene coordinate for each pixel, thereby enabling PnP-based pose estimation without an image database or SfM map. Several approaches [10, 32, 33] further accelerate the inference stage by directly regressing the camera pose from a 2D image, making them independent of the PnP+RANSAC stage. While these Absolute Pose Regression (APR) methods achieve fast inference, they trade localization accuracy for inference speed and require scene-specific training, thus limiting their scalability to real-world multi-scene applications [34, 30, 35]. Recent works have attempted to overcome these limitations by accelerating per-scene training to a matter of minutes [30, 31, 36] or by optimizing a single model that can handle multiple scenes [37–39].

NeRF-based methods were initially introduced to bridge the gap between efficiency and accuracy in APR, either by augmenting training data via novel view synthesis [40, 41] or by aligning query features with implicit features [42, 43] from NeRF [44–46]. Beyond utilizing it for APR, several works leverage rendered depth and features from NeRF to reintroduce PnP-based pose estimation, achieving strong accuracy with compact maps [43, 47, 48].

GS-based methods have further advanced localization accuracy upon explicit 3D Gaussian representations. Leveraging the real-time, high-fidelity photometric rendering of GS, fast and robust pose refinement has become readily available [49–52]. In parallel, as 3D feature fields [4, 6, 5] embed preprocessed 3D features aligned with 2D ones, GS-based feature fields (GSFFs) naturally support feature rendering-based pose refinement. Notably, STDLoc [9] treats GSFFs as the predefined 3D map itself, employing Gaussians not only for feature rendering but directly for 2D-3D feature matching, thereby establishing an end-to-end localization pipeline upon GSFFs.

However, this entails hours of per-scene training for feature fields, and the resulting feature fields are not inherently suited for 2D-3D matching. We attribute this to redundant Gaussians and ambiguous pixel-Gaussian correspondences induced by photometric optimization, both undermining direct 2D-3D matching. In contrast, we construct localization-specialized GSFFs through composition weight-based feature lifting and Mixture-of-Gaussians-based splitting, explicitly decomposing Gaussian primitives to alleviate many-to-one matching ambiguity without per-scene training for feature fields.

3 Method

3.1 Preliminary

3D Gaussian Splatting (3DGS) [3] extends 3D point clouds into an explicit set of 3D Gaussian primitives $\mathcal{G} = \{g_n\}_{n=1}^{N_g}$, where N_g denotes the number of Gaussians, supporting efficient differentiable volumetric rendering. Each 3D Gaussian is parameterized by a mean μ, an anisotropic covariance $\Sigma = RSS^\top R^\top$ (with R and S representing rotation and scale, respectively), an opacity α, and spherical harmonic (SH) color coefficients $\mathbf{c}$. For the j-th pixel in i-th image I_{ij}, the rendered color is computed as $\hat{I}_{ij} = \sum_n w(g_n, I_{ij})\, \mathbf{c}_n$, where $w(g_n, I_{ij})$ denotes the weight assigned to each Gaussian during the rasterization process, reflecting both the accumulated transmittance along the ray and its opacity. We hereafter refer to this term as the composition weight.

Building upon the photometric 3DGS framework, GS-based Feature Fields (GSFFs) [5, 6] augment each Gaussian g_n with a feature vector $\mathbf{z}_n$. For localization, GSFFs serve as a 3D map in which each Gaussian acts as a 3D point with $\mathbf{z}_n$ as its descriptor. The camera pose of a query image is then estimated by PnP+RANSAC [1, 2] on 2D-3D matches obtained via feature similarity. To construct such GSFFs, a common formulation is to optimize $\mathbf{z}_n$ by supervising the rendered feature map $\hat{\mathbf{f}}_{ij} = \sum_n w(g_n, I_{ij})\, \mathbf{z}_n$ against the corresponding pixel-level features $\mathbf{f}_{ij}$ extracted by a 2D image encoder, so that the resulting feature field is aligned with the encoder's feature space. However, we argue that these approaches are less effective for direct matching, as rendering-based optimization causes feature blurring by distributing gradients over hundreds of Gaussians per ray [13, 14], thereby

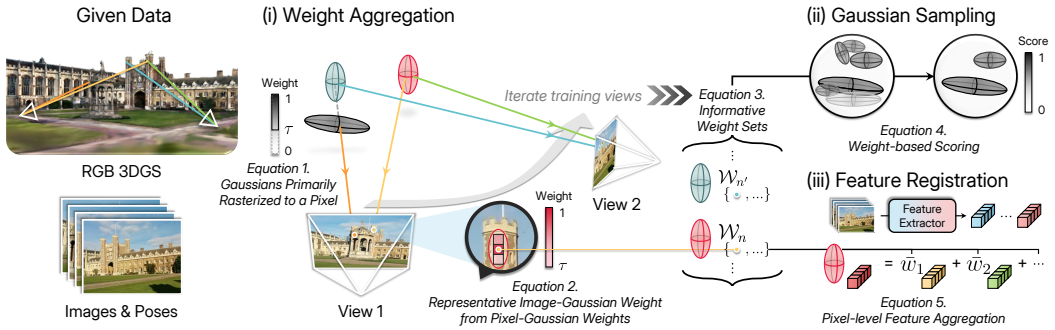


Figure 2: **PlugGS-Loc pipeline.** The figure illustrates how to construct GSFFs for localization when pretrained RGB 3DGS and training views are given. (i) For each view I_i, we render and identify Gaussians primarily rasterized above the weight threshold τ in Eq. 1 for each pixel, and select a representative image-Gaussian weight among them for memory efficiency as in Eq. 2. After iterating over all training views, the resulting informative weight sets defined in Eq. 3 serve as the core of both (ii) Gaussian Sampling and (iii) Feature Registration. The former selects essential Gaussians, while the latter assigns multi-view aggregated features to those Gaussians. Note that the entire process is streamlined and thus can be completed within a minute per scene.

weakening 2D-3D correspondences for PnP. Accordingly, we adopt a direct feature lifting pipeline inspired by recent open-vocabulary semantic segmentation works [14, 53], but tailor it for localization.

3.2 PlugGS-Loc

Weight aggregation. To identify Gaussians that are suitable for 2D-3D matching, we first gather the set of Gaussians primarily rasterized to the corresponding pixel I_{ij} by applying a threshold τ on their composition weights:

$$\hat{\mathcal{G}}_{ij} = \{g_n \in \mathcal{G} \mid w(g_n, I_{ij}) \geq \tau\}. \tag{1}$$

Intuitively, this preserves Gaussians that contribute more than τ (e.g., $\tau = 0.1 \rightarrow 10\%$) of the total composition weight for the pixel I_{ij}. However, storing all of them requires a huge amount of memory, which is infeasible in practice. Instead, we exploit only the maximum composition weight over all pixels in the view, forming one representative weight for each image-Gaussian pair. For each Gaussian g_n and training image I_i, it is defined as:

$$\tilde{w}(g_n, I_i) := \begin{cases} \max\limits_{j} \left(w(g_n, I_{ij})\right), & \text{if } \cup \hat{\mathcal{G}}_{ij} \neq \emptyset, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

In essence, $\tilde{w}(g_n, I_i)$ captures the strongest pixel-Gaussian contribution in a view I_i as illustrated in Fig. 2. We then iterate all training images to collect multi-view contributions, finally obtaining a compact and informative weight set for each Gaussian as:

$$\mathcal{W}_n = \{\tilde{w}(g_n, I_i) \mid \tilde{w}(g_n, I_i) > 0\}. \tag{3}$$

$\{\mathcal{W}_n\}_{n=1}^{N_g}$ lay the foundation for both Gaussian sampling and feature registration, supporting the streamlined GSFFs construction pipeline. Note that only non-zero weights are considered, so Gaussians with $\mathcal{W}_n = \emptyset$ are excluded in the next step.

Gaussian sampling. As exhaustive 2D-3D matching becomes computationally expensive with numerous Gaussians (and as many of them are redundant due to photometric optimization), it is essential to prune superfluous ones. Following STDLoc [9], we define each local 3D region by randomly sampling anchor Gaussians to cover the scene and grouping their k-nearest neighbors in terms of the L2 distance between Gaussian centers. Then, we score each Gaussian's importance and retain only the one with the highest score within each local region. Unlike STDLoc [9], we utilize the informative weight set $\mathcal{W}_n$ to measure the importance of each Gaussian as

$$\psi(g_n) = \frac{1}{|\mathcal{W}_n|} \sum_{\tilde{w} \in \mathcal{W}_n} \tilde{w}, \tag{4}$$

where $|\mathcal{W}_n|$ denotes the cardinality of $\mathcal{W}_n$. This weight-based criterion favors Gaussians with high average contributions over views with non-zero weights. Since it measures the average contribution conditioned on visibility, a Gaussian strongly observed in only a few views could dominate one moderately observed in many views. Nevertheless, under dense training views, this scoring and sampling within local regions empirically yield multi-view supported Gaussians distributed across the scene, supporting robust matching in novel views while maintaining a compact map.

Feature registration. Instead of rendering-based optimized $\mathbf{z}_n$, we adopt $\bar{\mathbf{z}}_n$, which is computed as a weighted sum of the corresponding pixel features as follows:

$$\bar{\mathbf{z}}_n = \sum_{(i,j^*)\in\mathcal{P}_n} \bar{w}_{ij^*} \bar{\mathbf{f}}_{ij^*}, \tag{5}$$

where $\mathcal{P}_n = \{(i, j^*) \mid w(g_n, I_{ij^*}) = \tilde{w}(g_n, I_i)\}$ is a set of pixel indices that correspond to $\mathcal{W}_n$, $\bar{w}$ denotes the softmax-normalized weight within $\mathcal{W}_n$, and $\bar{\mathbf{f}}$ denotes pixel-level image feature $\mathbf{f}$ normalized along the channel dimension. It enables us to obtain multi-view aggregated representations, finally constructing GSFFs within a minute. In summary, PlugGS-Loc efficiently identifies the most informative Gaussians and associates them with features exhibiting strong pixel-Gaussian correspondence, resulting in robust 2D-3D matching for localization.

3.3 SplitGS-Loc

SplitGS-Loc is built on the key insight that photometrically optimized Gaussians are ill-suited for 2D-3D matching since their volumetric extent induces a structural mismatch with PnP. A Gaussian can be associated with several pixels, but since PnP requires point-to-point correspondences as input, many-to-one matches inevitably accumulate reprojection errors and thus produce suboptimal results. To address the issue, we introduce a Mixture-of-Gaussians-based splitting strategy that decomposes each Gaussian along its major axis, the direction most prone to ambiguity.

Mixture of Gaussians and 3DGS. A Gaussian distribution, omitting the index n for notational simplicity, can be formulated as $p(x) \sim \mathcal{N}(\mu, \Sigma)$, which extends to a mixture as

$$p(x) = \sum_{k=1}^{K} \lambda_k \mathcal{N}(x|\mu_k, \Sigma_k), \tag{6}$$

where the components form a linear combination with coefficients λ_k satisfying $\sum_k \lambda_k = 1$. In 3DGS, this corresponds to distributing a Gaussian's opacity α across its components such that the k-th child carries opacity $\lambda_k\alpha$. The covariance, as introduced in Sec. 3.1, is given by $\Sigma = RSS^\top R^\top$, with the scale matrix $S = \mathrm{diag}(s_x, s_y, s_z)$ describing the radii of the Gaussian ellipsoid along the x-, y-, and z-axes, respectively. Since a multivariate Gaussian factorizes into independent components along its principal axes, we hereafter consider a mixture of 1D Gaussians.

Mixture-of-Gaussians-based splitting. In order to reduce the pixel-Gaussian matching ambiguity, particularly along the major direction with the highest ambiguity tendency, we decompose each 3D Gaussian by symmetrically adding two side components along its major axis. Formally, in the canonical space and considering only the major axis x such that $\tilde{p}(x) \sim \mathcal{N}(0, s^2)$, $\tilde{p}(x)$ can be represented as a mixture of Gaussians:

$$\tilde{p}(x) \approx \tilde{q}(x) = \lambda q_-(x) + \lambda_\mathrm{o} q_\mathrm{o}(x) + \lambda q_+(x), \tag{7}$$

where s denotes the standard deviation (*i.e.*, scale) along the selected major axis, $2\lambda + \lambda_\mathrm{o} = 1$, and each component is defined as:

$$\begin{aligned} q_-(x) &\sim \mathcal{N}(-\beta s,\ \sigma^2), \\ q_\mathrm{o}(x) &\sim \mathcal{N}(0,\ \sigma_\mathrm{o}^2), \\ q_+(x) &\sim \mathcal{N}(+\beta s,\ \sigma^2), \end{aligned} \tag{8}$$

where $\beta > 0$ controls the side offsets, and $\sigma^2, \sigma_\mathrm{o}^2$ are the component variances.

Based on this formulation, the parameters are derived by matching the first four moments of $\tilde{p}(x)$ and $\tilde{q}(x)$. Moment matching determines $\lambda_\mathrm{o} = \frac{2}{3}$, $\lambda = \frac{1}{6}$, and $\sigma^2 = \sigma_\mathrm{o}^2 = (1 - \frac{\beta^2}{3})s^2$ for any $\beta \in (0, \sqrt{3})$.

Table 1: **Localization results on Cambridge Landmarks [10].** We report median translation and rotation errors. For all GS-based methods, mapping time includes RGB 3DGS [3] training time (about 6min), and map size includes per-scene optimized networks and feature fields on top of 3DGS (∼80MB). $^{+}$ denotes the version with iterative camera pose refinement using GS-CPR [49], and † denotes methods that need additional training to adapt the representation for localization per scene.

		Mapping Time	Pose Refine	Size (MB)	Great Court	King's College	Old Hospital	Shop Facade	St. M. Church	Average (cm/°)
FM	AS (SIFT) [17]	0		∼200	24/0.13	13/0.22	20/0.36	4/0.21	8/0.25	13.8/0.23
	HLoc (SP+SG) [19, 26]	0		∼800	17.7/0.11	11.0/0.20	15.1/0.31	4.2/0.20	7.0/0.22	11.0/0.21
	PixLoc [21]	0	✓	∼600	30/0.14	14/0.24	16/0.32	5/0.23	10/0.34	15.0/0.25
SCR	DSAC* [29]	15h		28	34/0.2	18/0.3	21/0.4	5/0.3	15/0.6	19/0.4
	ACE [30]	5min		4	43/0.2	28/0.4	31/0.6	5/0.3	18/0.6	25/0.4
	GLACE [31]	20min		13	19/0.1	19/0.3	17/0.4	4/0.2	9/0.3	14/0.3
Neural Scene Representation†										
NeRF	CROSSFIRE [43]	15h	✓	50	N/A	47/0.7	43/0.7	20/1.2	39/1.4	N/A
	PNeRFLoc [47]	hrs	✓	∼50	81/0.25	24/0.29	28/0.37	6/0.27	40/0.55	35.8/0.35
	NeRFMatch [48]	hrs	✓	∼55	19.6/0.09	12.5/0.23	20.9/0.38	8.4/0.40	10.9/0.35	14.5/0.29
GS	GSplatLoc [7]	hrs	✓	>100	N/A	31/0.49	16/0.68	4/0.34	14/0.42	N/A
	GSFFs-PR [8]	hrs	✓	>100	N/A	17/0.26	18/0.36	4/0.25	8/0.26	N/A
	STDLoc [9]	2h		16	**13.1**/0.08	15.2/0.17	18.6/0.36	3.5/0.17	5.3/0.19	11.1/0.19
	STDLoc (PR) [9]	2h	✓	415	15.7/**0.06**	15.0/0.17	11.9/0.21	**3.0/0.13**	**4.7**/0.14	**10.1/0.14**
	GSVisLoc^{+} [49, 50]	hrs	✓	∼80	N/A	23/0.3	22/0.42	8/0.29	14/0.45	N/A
	RAP^{+} [49, 51]	hrs	✓	∼80	22/0.15	15/0.23	18/0.38	5/0.23	9/0.23	13.8/0.24
	GS-RelocNet [52]	hrs		>100	N/A	11/0.19	13/0.26	4/0.18	7/0.15	N/A
	GS-RelocNet (PR) [52]	hrs	✓	>100	N/A	**9/0.15**	**10/0.19**	**3**/0.15	5/**0.13**	N/A
Neural Scene Representation										
GS	GS-RelocNet (PR) [52]	6min	✓	∼80	N/A	**12**/0.18	**13/0.25**	5/0.19	7/0.20	N/A
	PlugGS-Loc (Ours)	7min		16	11.7/0.08	15.9/0.18	16.7/0.37	2.9/0.13	6.0/0.21	10.6/0.20
	SplitGS-Loc (Ours)	7min		97	**9.3/0.06**	14.1/**0.17**	13.8/0.29	**2.5/0.12**	**3.8/0.13**	**8.7/0.15**

Under this configuration, the side components are positioned along the ellipsoid's major axis at $\pm\beta s$, spanning a wider extent of the original distribution and thus reducing the pixel-Gaussian matching ambiguity. For the remaining axes, the child Gaussians follow the original distribution, and are finally transformed back to the original coordinate frame. Note that this process runs with one million Gaussians in under one second. A detailed derivation is provided in the appendix.

Integration of splitting strategy and PlugGS-Loc. Once splitting strategy is applied, the resulting child Gaussians are seamlessly integrated into the PlugGS-Loc to transform dense GS into localization-specialized GSFFs. Regarding Gaussian sampling, however, directly applying Eq. 4 to child Gaussians would undermine the one-to-one correspondences established by splitting. This is because multiple child Gaussians within the same local region would collapse into a single representative. To prevent this, we temporarily aggregate the child Gaussians only at the sampling stage, applying the new score aggregated into the original Gaussian, defined as:

$$\bar{\psi}(g_n) = \frac{1}{3}\big(\psi(g_{n_-}) + \psi(g_{n_\circ}) + \psi(g_{n_+})\big), \tag{9}$$

where g_{n_-}, $g_{n_\circ}$, and g_{n_+} are child Gaussians split from the original Gaussian g_n. Then, sampling and scoring are performed based on the original Gaussians' centers $\{\mu_n\}_{n=1}^{N_g}$ and aggregated scores $\{\bar{\psi}(g_n)\}_{n=1}^{N_g}$, and corresponding child Gaussians of the selected parents are finally preserved. Subsequently, they are independently paired with different pixel features during feature registration; therefore, the salient region once represented by a single descriptor is described by multiple fine-grained ones. As a result, SplitGS-Loc yields compact yet more discriminative GSFFs, mitigating the pixel-Gaussian matching ambiguity, thus enabling accurate matching and stable PnP convergence.

4 Experiment

4.1 Experimental setup

Datasets. We conducted experiments on Cambridge Landmarks [10] and 7Scenes [15] datasets. Cambridge Landmarks consists of five outdoor scenes, with scene scales ranging from 875m^2 to

Table 2: **Localization results on 7Scenes [15].** We report median translation and rotation errors. For all GS-based methods, mapping time includes RGB 3DGS [3] training time (about 6min), and map size includes per-scene optimized networks and feature fields on top of 3DGS (∼175MB). $^+$ denotes the version with iterative camera pose refinement using GS-CPR [49], and † denotes methods that need additional training to adapt the representation for localization per scene.

		Mapping Time	Pose Refine	Size (MB)	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Average (cm/°)
FM	AS (SIFT) [17]	0		∼200	3/0.87	2/1.01	1/0.82	4/1.15	7/1.69	5/1.72	4/1.01	3.7/1.18
	HLoc (SP+SG) [19, 26]	0		∼2000	2.4/0.84	2.3/0.91	1.1/0.77	3.1/0.92	4.9/1.30	4.2/1.39	5.1/1.41	3.3/1.08
	PixLoc [21]	0	✓	∼1000	2/0.80	2/0.73	1/0.82	3/0.82	4/1.21	3/1.20	5/1.30	2.9/0.98
SCR	DSAC* [29]	15h		28	0.5/0.17	0.8/0.29	0.5/0.34	1.2/0.35	1.2/0.29	0.7/0.21	2.7/0.78	1.1/0.35
	ACE [30]	5min		4	0.6/0.18	0.8/0.33	0.5/0.33	1.1/0.29	1.1/0.22	0.8/0.21	2.9/0.81	1.1/0.34
Neural Scene Representation†												
NeRF	CROSSFIRE [43]	5h	✓	50	1/0.4	5/1.9	3/2.3	5/1.6	3/0.8	2/0.8	12/1.9	4.4/1.38
	PNeRFLoc [47]	hrs	✓	∼50	2/0.80	2/0.88	1/0.83	3/1.05	6/1.51	5/1.54	32/5.73	7.3/1.76
	NeRFMatch [48]	hrs	✓	∼55	1.0/0.30	1.1/0.41	1.3/0.92	3.1/0.87	2.2/0.60	1.0/0.28	9.3/1.74	2.7/0.73
GS	GSplatLoc [7]	hrs	✓	>200	0.4/0.16	1.0/0.32	1.1/0.62	1.9/0.4	1.8/0.35	2.7/0.55	8.8/2.34	2.5/0.68
	GSFFs-PR [8]	hrs	✓	>200	0.4/0.19	0.6/0.26	0.5/0.36	1.0/0.31	1.3/0.38	0.6/0.23	25/0.63	4.2/0.34
	STDLoc [9]	2h		16	1.0/0.21	0.9/0.30	0.6/0.30	1.4/0.30	1.2/0.22	1.4/0.23	2.7/0.71	1.3/0.32
	STDLoc (PR) [9]	2h	✓	888	0.5/0.15	0.6/0.24	0.5/**0.26**	0.9/0.24	0.9/0.21	0.6/0.19	1.4/0.41	0.8/0.24
	GSVisLoc$^+$ [49, 50]	hrs	✓	>200	0.4/0.13	0.6/0.24	0.5/0.34	1.0/0.26	0.9/0.21	0.7/0.18	4.7/0.96	1.3/0.33
	RAP$^+$ [49, 51]	hrs	✓	∼175	**0.3/0.11**	**0.5/0.21**	**0.4**/0.27	**0.6/0.16**	**0.8/0.20**	**0.5/0.12**	**1.1/0.32**	**0.6/0.20**
	GS-RelocNet [52]	hrs		>200	0.4/0.17	0.6/0.24	**0.4**/0.30	0.9/0.24	1.0/0.28	0.6/0.22	1.4/0.39	0.8/0.26
	GS-RelocNet (PR) [52]	hrs	✓	>200	0.4/0.15	0.6/**0.21**	**0.4/0.26**	0.9/0.24	0.9/0.21	0.6/0.18	1.3/0.35	0.7/0.23
Neural Scene Representation												
GS	GS-RelocNet [52]	6min		∼175	1.3/0.81	1.2/0.65	0.7/0.73	1.6/0.89	2.7/1.01	2.5/1.10	2.1/1.00	1.7/0.88
	GS-RelocNet (PR) [52]	6min	✓	∼175	1.0/0.72	1.0/0.64	0.6/0.70	1.4/0.82	2.2/0.93	2.0/1.02	1.9/0.92	1.4/0.82
	PlugGS-Loc (Ours)	7min		16	0.9/0.18	0.9/0.31	0.7/0.30	1.4/0.30	1.6/0.27	1.3/0.23	1.8/0.50	1.2/0.30
	SplitGS-Loc (Ours)	7min		88	**0.7/0.16**	**0.7/0.29**	**0.6/0.26**	**1.2/0.25**	**1.4/0.25**	**1.0/0.18**	**1.6/0.47**	**1.0/0.26**

7600m^2. In contrast, 7Scenes includes seven indoor scenes whose scales range from 1m^2 to 18m^2. For the ground-truth camera poses, we adopted the SfM-based annotations reported by [54], as the D-SLAM-based ground truths are unreliable for neural rendering [49].

Implementation details. We basically follow the setting of STDLoc [9]. For the weight aggregation, the weight threshold τ in Eq. 1 is fixed at 0.1 across all scenes to avoid per-scene optimization. We pre-filter superfluous Gaussians via proposed weight aggregation process before PlugGS-Loc and SplitGS-Loc to obtain a clean weight set. In SplitGS-Loc, the number of anchor Gaussians retained during sampling is doubled to ensure sufficient spatial coverage. The splitting factor β in Eq. 8 is set to 1.4 to extend the side components further along the major axis while preserving the approximation quality under the positive semi-definiteness constraint. For feature aggregation and inference, we use the same image feature extractor and PnP+RANSAC algorithm as STDLoc, SuperPoint [27] and PoseLib [55], respectively. All experiments are conducted on a single NVIDIA GeForce RTX 4090.

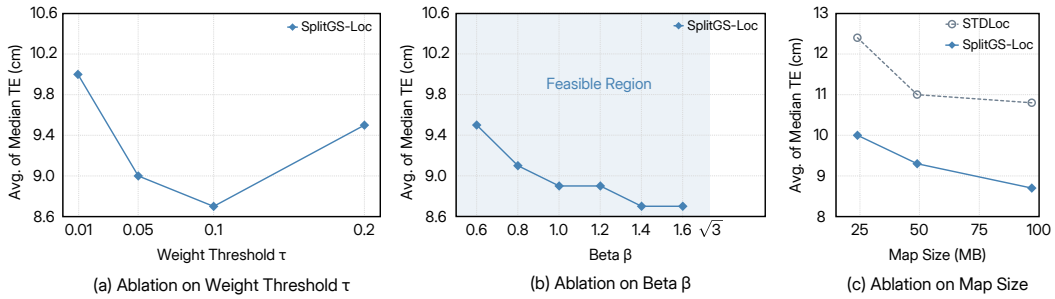
Metrics. Our GSFFs construction on top of a pre-trained 3DGS completes within a minute, where the reported mapping time additionally includes RGB 3DGS [3] training time (about 6min) shared across all GS-based methods for fair comparison. Mapping time and map sizes are estimated according to the data required at inference. Details are provided in the appendix.

4.2 Comparative results

Our proposed frameworks show superior performance on both indoor and outdoor scenes as shown in Tab. 2 and Tab. 1. Notably, SplitGS-Loc achieves remarkable performance on the largest scene, GreatCourt, without per-scene training for feature fields or pose refinement. These results demonstrate that accurate localization can be achieved without relying on heavy GSFFs or iterative refinement, moving toward a streamlined solution for real-world visual localization: smaller, faster, yet more precise.

4.3 Ablation studies

Weight threshold τ. We conduct an ablation on the weight threshold τ that determines the importance of each Gaussians, testing values of $0.01, 0.05, 0.1, 0.2$, as depicted in Fig. 3 (a). When the pruning is too mild, features that have weak pixel-Gaussians associations are aggregated together,



(a) Ablation on Weight Threshold τ (b) Ablation on Beta β (c) Ablation on Map Size

Figure 3: **Ablation on mapping hyperparameters.** We compare the average per-scene median translation error under varying weight thresholds τ, splitting factor β, and the map size on Cambridge Landmarks [10]. (a) Mild pruning causes feature blurring and aggressive pruning limits coverage of the scene, bringing suboptimal results. (b) Localization accuracy increases monotonically with β within the feasible region, suggesting that wider separation of child Gaussians more effectively alleviates pixel-Gaussian matching ambiguity. (c) SplitGS-Loc scales consistently beyond the saturation by effective densification at the level of detail.

Table 3: **Alternative methods.** We compare alternative designs for SplitGS-Loc on Cambridge Landmarks [10]. (a) Our GSFFs and sampling strategy bring complementary improvements without multi-stage training. (b) Replacing Gaussian attributes with position-only alternatives degrades accuracy.

(a) Ablation with STDLoc [9].

GSFFs	Sampling	Average (cm/°)
STDLoc	Cosine Similarity	13.1/0.22
STDLoc	Eq. 4	12.8/0.22
Ours	Cosine Similarity	9.6/0.17
Ours	Eq. 9	**8.7/0.15**

(b) Ablation on Gaussian Attributes.

Feature Aggregation	Densification	Average (cm/°)
Projection+Average	-	14.1/0.26
PlugGS-Loc (Ours)	-	10.6/0.20
Ours	NN Upsampling	10.0/0.18
SplitGS-Loc (Ours)	Eq. 7, Eq. 8	**8.7/0.15**

resulting in blurred representations and thus degraded localization accuracy. This reflects a limitation inherent to rendering-based GSFFs optimization as discussed in Sec. 3.1, where volumetric rendering disperses gradients across numerous Gaussians per ray and weakens feature discriminativeness. Conversely, when the pruning is too aggressive, the absolute number of remaining Gaussians and their spatial coverage become insufficient for stable pose estimation. Meanwhile, our methods are robust to this hyperparameter within a reasonable range, with 0.05 performing on par with 0.1.

Splitting factor β. Fig. 3 (b) illustrates the effect of the splitting factor β that controls the side offsets of child Gaussians. Localization accuracy improves monotonically as β increases within the feasible region ($\beta \in (0, \sqrt{3})$), where the positive semi-definiteness constraint is satisfied. This suggests that, once the moment matching approximation holds, placing child Gaussians further apart along the major axis more effectively reduces pixel-Gaussian matching ambiguity.

Component ablation. Tab. 3 (a) provides a separate ablation using STDLoc [9] as the baseline, independently swapping the GSFFs and the sampling strategy. Our GSFFs yields a significant accuracy improvement from 13.1cm/0.22° to 9.6cm/0.17°, demonstrating that the proposed framework is not only efficient but also more effective. In parallel, our weight-based sampling strategy brings additional consistent improvement, achieving the best performance of 8.7cm/0.15°. The result confirms that the composition weight acts as a unified criterion that simultaneously guides feature registration and Gaussian sampling, collectively enabling a more accurate and efficient GSFFs construction.

4.4 Analysis

Map capacity. Larger maps generally lead to better performance, similar to how dense reconstruction-based pipelines outperform sparse ones. Accordingly, the key questions are (i) how they perform under the same map capacity and (ii) how effectively the performance scales with increasing map size. To examine this, as illustrated in Fig. 3 (c), we compare direct 2D-3D matching

Table 4: **PnP convergence.** We investigate the correlation between PnP convergence and accuracy on Cambridge Landmarks [10], comparing STDLoc [9] and our SplitGS-Loc under efficient (a,c) and default (b,d) setting. By mitigating the pixel-Gaussian matching ambiguity, SplitGS-Loc produces fewer many-to-one inputs and more valid inliers, achieving the best accuracy with stable convergence.

	Direct 2D-3D Matching	RANSAC Min. Iter.	RANSAC Max. Iter.	# of PnP Many-to-one (↓)	# of PnP Final Inliers (↑)	Runtime (↓) (ms)	Average (↓) (cm / °)
(a)	STDLoc	10^2	10^3	313	362	54.4	12.8/0.21
(b)	STDLoc	10^3	10^5	313	367	314.4	11.2/0.20
(c)	SplitGS-Loc (Ours)	10^2	10^3	199	447	54.8	8.9/**0.15**
(d)	SplitGS-Loc (Ours)	10^3	10^5	199	450	215.5	**8.7/0.15**

pipelines using similar map sizes between STDLoc [9] and proposed SplitGS-Loc. For fairness, we exclude any scene-specific post-training and adjust the number of sampled Gaussians to approximate the same map size, using 3×2^{13}, 3×2^{14}, 3×2^{15} samples for STDLoc, and 2^{13}, 2^{14}, 2^{15} (one third) for SplitGS-Loc as each Gaussian is further divided into three child Gaussians. Fig. 3 (c) shows that merely increasing the number of sampled Gaussians faces an inherent limitation: the performance saturates. By contrast, our SplitGS-Loc exhibits sustained improvements as the map size increases, highlighting the efficacy of our splitting strategy.

Advantages of Gaussian attributes. We investigate the advantages of leveraging Gaussian attributes for localization. Tab. 3 (b) presents results when Gaussian attributes are replaced with geometry-only alternatives under the same pipeline. First, we replace our rasterization-based pipeline with Projection+Average. Our method filters strong pixel-Gaussian correspondences via composition weights and aggregates features through soft assignment, whereas Projection+Average directly projects Gaussian centers onto training views and uniformly averages the nearest keypoint descriptors. This leads to a significant median error increase from 10.6cm/0.20° to 14.1cm/0.26°; composition weights derived from opacity and transmittance naturally account for occlusion and visibility, whereas geometric projection alone cannot.

Besides, replacing our Gaussian Splitting with k-nearest neighbor upsampling ($k = 3$) results in 10.0cm/0.18°, which inserts new Gaussians at midpoint positions between each Gaussian and its nearest neighbors with linearly interpolated features, compared to 8.7cm/0.15° for SplitGS-Loc. This suggests that proposed Mixture-of-Gaussians-based splitting guided by scale enables more accurate densification. In summary, exploiting composition weights and volumetric attributes during mapping yields substantially more discriminative feature fields.

PnP convergence. Tab. 4 presents a PnP convergence analysis of direct 2D-3D matching-based methods under different RANSAC settings. While STDLoc (a, b) struggles to reach consensus due to redundant many-to-one matches, our SplitGS-Loc (c, d) mitigates the pixel-Gaussian matching ambiguity, reducing many-to-one inputs from 313 to 199. This contributes to obtaining more inliers, better accuracy, and faster convergence: from 314ms to 215ms under the same setting. This stable convergence in turn yields robustness under an efficient setting: while STDLoc suffers a notable accuracy drop from 11.2cm to 12.8 cm (b vs a), SplitGS-Loc remains stable about 8.7cm (d vs c), suggesting that its robustness stems from overall high-quality matches.

Limitation. While the proposed framework refines photometrically optimized Gaussians, it is not entirely independent of the quality of original Gaussians. As such, scenes with severe illumination changes or sparse trajectories over large-scale environments lie beyond what can be meaningfully examined with current GS techniques, as GS struggles to converge in such settings. This dependency is shared across neural scene representation-based localization, which has so far been validated only on small- to medium-scale scenes. Still, our conversion takes under a minute and applies to any explicit Gaussian formulation, so it can be readily plugged in whenever a well-trained large-scale 3DGS becomes available. Given the rapid progress of neural scene representation research, extending to such challenging real-world conditions would be an interesting direction for future work.

5 Conclusion

Recent years have seen significant progress in Gaussian Splatting-based representations, and their potential for visual localization continues to broaden. Yet photometrically optimized GSFFs are fundamentally ill-suited for 2D-3D matching, as redundant Gaussians and many-to-one pixel-Gaussian correspondences undermine both matching robustness and PnP convergence. This work has proposed SplitGS-Loc, which resolves these limitations effectively by exploiting Gaussian attributes for localization-specialized GSFFs construction, eliminating the need for per-scene multi-stage optimization. By disambiguating pixel-Gaussian correspondences, it ensures stable PnP convergence and sustained accuracy gains as the map scales, achieving state-of-the-art performance at a fraction of the cost that prior pipelines require. Beyond localization accuracy, this work reinforces long-standing goals in visual localization: efficiency and practicality.

Acknowledgements

This work was supported in part by MSIT/IITP (No. RS-2022-II220680, RS-2020-II201821, RS-2019-II190421, RS-2024-00459618, RS-2024-00360227, RS-2026-25549037, RS-2025-25442569), MSIT/NIPA (No. RS-2026-25621569), MSIT/NRF (No. RS-2024-00357729), and KNPA/KIPoT (No. RS-2025-25393280).

References

[1] 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] 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.

[3] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[4] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 19729–19739, 2023.

[5] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024.

[6] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024.

[7] Gennady Sidorov, Malik Mohrat, Denis Gridusov, Ruslan Rakhimov, and Sergey Kolyubin. Gsplatloc: Grounding keypoint descriptors into 3d gaussian splatting for improved visual localization. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12601–12607. IEEE, 2025.

[8] Maxime Pietrantoni, Gabriela Csurka, and Torsten Sattler. Gaussian splatting feature fields for (privacy-preserving) visual localization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1082–1092, 2025.

[9] Zhiwei Huang, Hailin Yu, Yichun Shentu, Jin Yuan, and Guofeng Zhang. From sparse to dense: Camera relocalization with scene-specific detector from feature gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27059–27069, 2025.

[10] 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.

[11] Ziyi Yang, Xinyu Gao, Yang-Tian Sun, Yi-Hua Huang, Xiaoyang Lyu, Wen Zhou, Shaohui Jiao, Xiaojuan Qi, and Xiaogang Jin. Spec-Gaussian: Anisotropic view-dependent appearance for 3D gaussian splatting. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/708e0d691a22212e1e373dc8779cbe53-Abstract-Conference.html.

[12] Alex Hanson, Allen Tu, Vasu Singla, Mayuka Jayawardhana, Matthias Zwicker, and Tom Goldstein. Pup 3d-gs: Principled uncertainty pruning for 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5949–5958, 2025.

[13] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, et al. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. *Advances in Neural Information Processing Systems*, 37:19114–19138, 2024.

[14] Kim Jun-Seong, GeonU Kim, Kim Yu-Ji, Yu-Chiang Frank Wang, Jaesung Choe, and Tae-Hyun Oh. Dr. splat: Directly referring 3d gaussian splatting via direct language embedding registration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14137–14146, 2025.

[15] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013.

[16] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. Location recognition using prioritized feature matching. In *European conference on computer vision*, pages 791–804. Springer, 2010.

[17] 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.

[18] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are large-scale 3d models really necessary for accurate visual localization? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1637–1646, 2017.

[19] 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.

[20] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Vincent Leroy, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar De Souza, and Gabriela Csurka. Robust image retrieval-based visual localization using kapture. *arXiv preprint arXiv:2007.13867*, 2020.

[21] 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.

[22] Khang Truong Giang, Soohwan Song, and Sungho Jo. Learning to produce semi-dense correspondences for visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19468–19478, 2024.

[23] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1808–1817, 2015.

[24] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[25] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[26] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[27] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.

[28] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4654–4662, 2018.

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

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

[31] 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.

[32] 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.

[33] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2616–2625, 2018.

[34] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3302–3312, 2019.

[35] Siyan Dong, Shuzhe Wang, Shaohui Liu, Lulu Cai, Qingnan Fan, Juho Kannala, and Yanchao Yang. Reloc3r: Large-scale training of relative camera pose regression for generalizable, fast, and accurate visual localization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16739–16752, 2025.

[36] 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.

[37] Yoli Shavit, Ron Ferens, and Yosi Keller. Coarse-to-fine multi-scene pose regression with transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[38] Tao Xie, Zhiqiang Jiang, Shuozhan Li, Yukun Zhang, Kun Dai, Ke Wang, Ruifeng Li, and Lijun Zhao. Ufvl-net: A unified framework for visual localization across multiple indoor scenes. *IEEE Transactions on Instrumentation and Measurement*, 72:1–16, 2023.

[39] Miso Lee, Jihwan Kim, and Jae-Pil Heo. Activating self-attention for multi-scene absolute pose regression. *Advances in Neural Information Processing Systems*, 37:38508–38529, 2024.

[40] Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-posenet: Absolute pose regression with photometric consistency. In *2021 International Conference on 3D Vision (3DV)*, pages 1175–1185. IEEE, 2021.

[41] 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. PMLR, 2022.

[42] 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, 2022.

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

[44] 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.

[45] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7210–7219, 2021.

[46] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

[47] Boming Zhao, Luwei Yang, Mao Mao, Hujun Bao, and Zhaopeng Cui. Pnerfloc: Visual localization with point-based neural radiance fields. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 7450–7459, 2024.

[48] Qunjie Zhou, Maxim Maximov, Or Litany, and Laura Leal-Taixé. The nerfect match: Exploring nerf features for visual localization. In *European Conference on Computer Vision*, pages 108–127, 2024.

[49] 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. URL `https://openreview.net/forum?id=mP7uV59iJM`.

[50] Fadi Khatib, Dror Moran, Guy Trostianetsky, Yoni Kasten, Meirav Galun, and Ronen Basri. Generalizable visual localization for gaussian splatting scene representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2025.

[51] Sihang Li, Siqi Tan, Bowen Chang, Jing Zhang, Chen Feng, and Yiming Li. Adversarial exploitation of data diversity improves visual localization. In *International Conference on Computer Vision (ICCV)*, 2025.

[52] Junyi Wang, Yuze Wang, Wantong Duan, Meng Wang, and Yue Qi. 3d gaussian splatting based scene-independent relocalization with unidirectional and bidirectional feature fusion. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=ewgZItWaHh`.

[53] Juliette Marrie, Romain Ménégaux, Michael Arbel, Diane Larlus, and Julien Mairal. Ludvig: Learning-free uplifting of 2d visual features to gaussian splatting scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7440–7450, 2025.

[54] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the Limits of Pseudo Ground Truth in Visual Camera Re-Localisation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[55] Viktor Larsson and contributors. PoseLib - Minimal Solvers for Camera Pose Estimation, 2020. URL `https://github.com/vlarsson/PoseLib`.

[56] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pages 71–91. Springer, 2024.

[57] Ron Ferens and Yosi Keller. Hyperpose: Hypernetwork-infused camera pose localization and an extended cambridge landmarks dataset. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11547–11557. IEEE, 2025.

A Derivation of the Splitting Parameters

This section provides a detailed derivation of the mixture parameters obtained by matching the first four moments. As mentioned in the main paper, we define a parent 1D Gaussian distribution in the canonical space as $\tilde{p}(x) \sim \mathcal{N}(0, s^2)$, and approximate it by a symmetric three-component mixture $\tilde{q}(x)$ as:

$$\begin{aligned} \tilde{p}(x) \approx \tilde{q}(x) = \lambda q_-(x) + \lambda_{\mathrm{o}} q_{\mathrm{o}}(x) + \lambda q_+(x), \\ q_-(x) \sim \mathcal{N}(-\beta s,\ \sigma^2), \\ q_{\mathrm{o}}(x) \sim \mathcal{N}(0,\ \sigma_{\mathrm{o}}^2), \\ q_+(x) \sim \mathcal{N}(+\beta s,\ \sigma^2). \end{aligned}$$

Here $\lambda, \lambda_{\mathrm{o}} \in (0, 1)$ are mixture weights satisfying $2\lambda + \lambda_{\mathrm{o}} = 1$, and $\beta > 0$ controls the side offsets.

Balanced Splitting. The purpose of splitting is to densify the 3DGS such that, after rendering, pixel-Gaussian associations become more localized and less ambiguous. From this perspective, an important requirement for the approximation $\tilde{q}(x)$ is that the three components $q_-(x)$, $q_{\mathrm{o}}(x)$, and $q_+(x)$ share the spatial extent of the original density in a balanced manner. If the central component $q_{\mathrm{o}}(x)$ were given a substantially larger variance than the side components, most of the probability mass would remain concentrated around the original mean. Then, the side components would contribute negligibly and tend to be pruned by the subsequent weight aggregation process. Conversely, if the side components $q_-(x)$ and $q_+(x)$ were overly spread, the mixture would deviate from the original Gaussian and degrade the approximation quality. To avoid these issues, we first impose $\sigma_{\mathrm{o}} = \sigma$, thereby reducing the degrees of freedom and biasing the solution toward our final goal, fine-grained localization.

Moment Matching. Since both distributions are symmetric around the origin, all odd-order moments are zero. In the following, we therefore describe how to match the second and fourth moments. For a univariate Gaussian random variable $Z \sim \mathcal{N}(m, v)$, the second and fourth moments are

$$\mathbb{E}[Z^2] = v + m^2, \ \ \mathbb{E}[Z^4] = 3v^2 + 6vm^2 + m^4. \tag{10}$$

To match the second-order moments of $\tilde{p}(x)$ and $\tilde{q}(x)$, the following constraint is imposed:

$$\mathbb{E}_{\tilde{p}}[x^2] = \mathbb{E}_{\tilde{q}}[x^2]. \tag{11}$$

Since $\tilde{p}(x) \sim \mathcal{N}(0, s^2)$, we have

$$\mathbb{E}_{\tilde{p}}[x^2] = s^2. \tag{12}$$

For the mixture $\tilde{q}(x)$, the second moment can be written as

$$\begin{aligned} \mathbb{E}_{\tilde{q}}[x^2] &= \lambda\, \mathbb{E}_{q_-}[x^2] + \lambda_{\mathrm{o}}\, \mathbb{E}_{q_{\mathrm{o}}}[x^2] + \lambda\, \mathbb{E}_{q_+}[x^2] \\ &= 2\lambda(\sigma^2 + \beta^2 s^2) + \lambda_{\mathrm{o}} \sigma^2. \end{aligned} \tag{13}$$

Substituting Eq. 12 and Eq. 13 into Eq. 11 yields

$$s^2 = 2\lambda(\sigma^2 + \beta^2 s^2) + \lambda_{\mathrm{o}} \sigma^2. \tag{14}$$

In parallel, the fourth-order moment matching between $\tilde{p}(x)$ and $\tilde{q}(x)$ is enforced by requiring

$$\mathbb{E}_{\tilde{p}}[x^4] = \mathbb{E}_{\tilde{q}}[x^4]. \tag{15}$$

The fourth moment of $\tilde{p}(x)$ is

$$\mathbb{E}_{\tilde{p}}[x^4] = 3s^4, \tag{16}$$

and for $\tilde{q}(x)$, we have

$$\begin{aligned} \mathbb{E}_{\tilde{q}}[x^4] &= \lambda\, \mathbb{E}_{q_-}[x^4] + \lambda_{\mathrm{o}}\, \mathbb{E}_{q_{\mathrm{o}}}[x^4] + \lambda\, \mathbb{E}_{q_+}[x^4] \\ &= 2\lambda\big(3\sigma^4 + 6\sigma^2(\beta s)^2 + (\beta s)^4\big) + \lambda_{\mathrm{o}}\big(3\sigma^4\big). \end{aligned} \tag{17}$$

From Eq. 16 and Eq. 17 together with Eq. 15,

$$3s^4 = 2\lambda\big(3\sigma^4 + 6\sigma^2(\beta s)^2 + (\beta s)^4\big) + \lambda_{\mathrm{o}}\big(3\sigma^4\big). \tag{18}$$

For simplicity, we introduce the dimensionless variable $U = \sigma^2/s^2$. Using $2\lambda + \lambda_o = 1$, the second-order constraint in Eq. 14 then becomes

$$(1 - \lambda_o)(U + \beta^2) + \lambda_o U = 1, \tag{19}$$

and the fourth-order constraint in Eq. 18 becomes

$$(1 - \lambda_o)\big(U^2 + 2U\beta^2 + \frac{\beta^4}{3}\big) + \lambda_o U^2 = 1. \tag{20}$$

Solving Eq. 19 for U gives

$$U = 1 - (1 - \lambda_o)\beta^2. \tag{21}$$

We then expand the left-hand side of Eq. 20 as

$$(1 - \lambda_o)\big((U + \beta^2)^2 - \frac{2}{3}\beta^4\big) + (\lambda_o U^2 - 1) = 0, \tag{22}$$

and substitute Eq. 21 into the left and right terms of Eq. 22 as

$$\begin{aligned} &(1 - \lambda_o)\big((U + \beta^2)^2 - \frac{2}{3}\beta^4\big) \\ =&(1 - \lambda_o)(1 + 2\lambda_o\beta^2 + \lambda_o^2\beta^4 - \frac{2}{3}\beta^4), \end{aligned} \tag{23}$$

and

$$\begin{aligned} &\lambda_o U^2 - 1 \\ =&\lambda_o\big(1 - 2(1 - \lambda_o)\beta^2 + (1 - \lambda_o)^2\beta^4\big) - 1 \\ =&(1 - \lambda_o)(-1 - 2\lambda_o\beta^2 + \lambda_o\beta^4 - \lambda_o^2\beta^4), \end{aligned} \tag{24}$$

respectively. Combining Eq. 23 and Eq. 24 yields

$$(1 - \lambda_o)(-\frac{2}{3}\beta^4 + \lambda_o\beta^4) = 0, \tag{25}$$

and finally we obtain a polynomial in λ_o,

$$\frac{\beta^4}{3}(3\lambda_o - 2)(\lambda_o - 1) = 0. \tag{26}$$

The $\lambda_o = 1$ corresponds to the degenerate case $\lambda = 0$, where the side components vanish and no splitting occurs. We therefore take $\lambda_o = 2/3$, and $U = \sigma^2/s^2 = 1 - \beta^2/3$. The covariance matrix is required to be positive semi-definite, which in our setting is equivalent to $\sigma^2 > 0$, *i.e.*, $1 - \beta^2/3 > 0$ and hence $\beta \in (0, \sqrt{3})$. Under this feasible range, we set $\beta = 1.4$, which places the side components further apart along the major axis while satisfying the positive semi-definiteness constraint, as validated by the ablation study in the main paper.

B Mapping time

Per-stage Processing Time. Tab. A1 reports the processing time of each stage in SplitGS-Loc, including Gaussian splitting, pre-filtering, weight aggregation, Gaussian sampling, and feature registration. Each stage is averaged across the Cambridge Landmarks [10]. If image features are precomputed, the feature registration cost can be further reduced; however, we do not account for this optimization, assuming only raw datasets are provided.

Table A1: **Per-stage processing time.**

Stage	Time (s)
Gaussian Splitting	0.4
Pre-filtering	10.6
Weight Aggregation	9.8
Gaussian Sampling	10.2
Feature Registration	33.4
Total	64.4

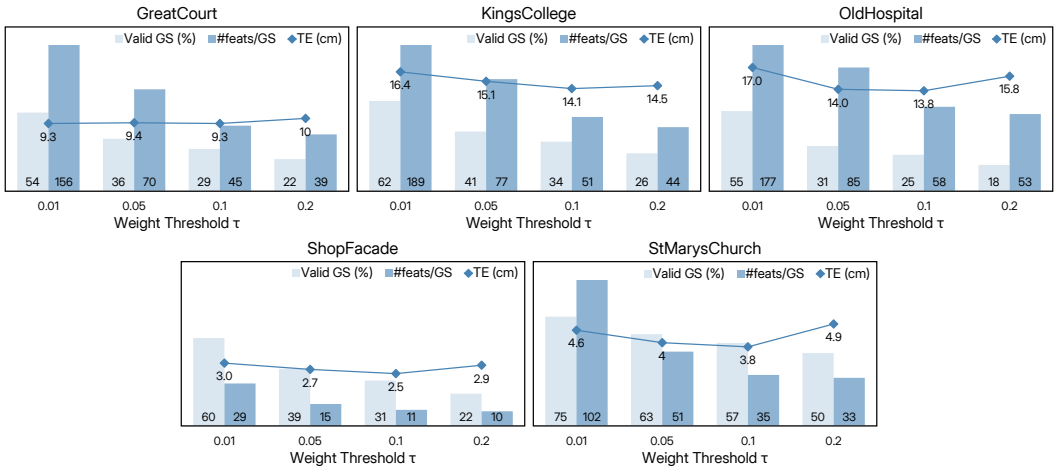


Figure A1: **Analysis of weight thresholding in SplitGS-Loc.**

Mapping Time. We set a SfM output as a zero across all methods and measure additional per-scene training (or post-processing) time. Accordingly, for all GS-based methods, the mapping time includes the time required to train the RGB 3DGS [3] (approximately 6 minutes per scene), even though it is not trained exclusively for localization. Although we make considerable effort to ensure a fair comparison across methods, achieving a fully fair comparison of mapping times is difficult in practice, since the mapping times of each method are highly sensitive to hardware, implementation details, and training schedules. For this reason, when a paper explicitly reports mapping times, we prioritize those published numbers [30, 43, 9].

For methods that employ a scene-specific model for pose initialization before pose refinement, we treat the training time of the initialization module as the per-scene mapping time. This is aligned with the protocol adopted in CROSSFIRE [43], which first introduced the neural scene representation-based relocalization, where the training time of the absolute pose regression model is included in the overall mapping cost. This also supports a fair comparison with direct 2D-3D matching pipelines such as STDLoc [9] and ours, which do not rely on any pose priors. For the remaining works, which provide no evidence of training completed in under an hour but involve multiple optimization epochs, we estimate their per-scene mapping process likely requires several hours.

Map size. Map sizes for scene coordinate regression and feature matching-based methods are referenced from ACE [30] and GLACE [31], while those for NeRF-based methods are standardized following the specifications in CROSSFIRE [43] and NeRFMatch [48]. For 3DGS-based methods, the map size is estimated approximately according to the data required at inference time: (i) if photometric rendering is required, the full RGB 3DGS is counted, (ii) if feature rendering is required, the full GSFFs are included, and (iii) for direct 2D-3D matching-based methods, only the 3D points and corresponding features are considered.

C Quantitative Analysis

This section extends the ablation on the weight threshold in Sec. 4.3 by providing a detailed analysis of how the primary Gaussian selection in Eq. 1 transforms a full photometric GS into localization-specialized GSFFs. Fig. A1 reports three statistics from the weight threshold τ: (i) the ratio of Gaussians whose informative weight set $\mathcal{W}_n$ (Eq. 3) is non-empty (Valid GS), (ii) the average number of multi-view features aggregated per valid Gaussian, *i.e.*, the average cardinality of the non-empty sets (feats/GS), and (iii) the median translation errors.

Although the scenes differ in terms of trajectories as well as environmental changes, we observe several trends that remain broadly consistent across scenes. First, even when preserving Gaussians that contribute at least 1% to any pixel ($\tau = 0.01$), about 40% of the Gaussians are excluded across the scenes. This result indicates that the photometric GS contains substantial redundancy from the perspective of solving the matching task, which requires strong correspondences.

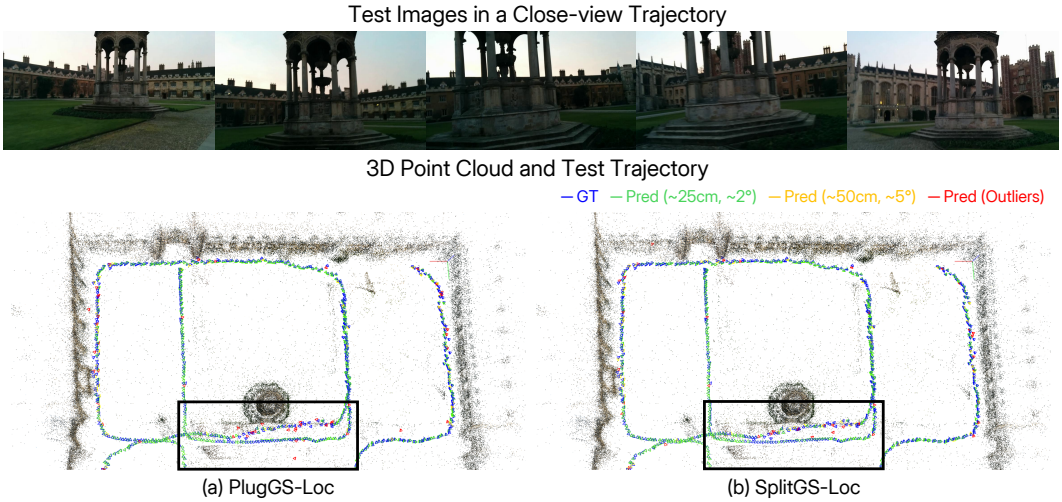


Figure A2: **Localization results of PlugGS-Loc and SplitGS-Loc on GreatCourt.**

Subsequently, we validate that simply aggregating a large number of multi-view features into each Gaussian does not directly guarantee improved localization accuracy in most scenes. Rather, as discussed in the main paper, maintaining a sufficient valid Gaussian ratio, *i.e.*, adequate spatial coverage of the scene, is critical. Finally, despite the diversity across scenes, the effective range of τ values does not vary dramatically. This suggests that the proposed composition weight-based GSFFs construction framework can be applied to new scenes without per-scene tuning of the weight threshold.

D Qualitative Analysis

Fig. A2 presents localization results on GreatCourt, the largest and most challenging outdoor scene in the Cambridge Landmarks dataset [10]. We color each predicted camera pose according to its translation and rotation errors, following common recall thresholds in large-scale visual localization: green for poses within $(25\text{cm}, 2^\circ)$, yellow for poses within $(50\text{cm}, 5^\circ)$, and red otherwise. Without any training for feature fields, PlugGS-Loc produces stable pose predictions along most of the trajectory. Nevertheless, it becomes less reliable when the camera observes the foreground at very close range, as shown in Fig. A2. This originates from pixel-Gaussian matching ambiguity discussed in Fig. 1. When photometrically optimized Gaussians are directly used for localization, those close to the camera tend to cover larger image regions and thus become associated with many pixels, exacerbating many-to-one pixel-Gaussian correspondences and destabilizing PnP convergence. Adopting SplitGS-Loc as introduced in Sec. 3.3 stabilizes pose estimation for these challenging views, as illustrated in Fig. A2.

E Pose Refinement

While SplitGS-Loc directly estimates the camera pose through 2D-3D matching using a compact feature map, its output can also serve as an initial pose for existing refinement methods. To examine the complementarity of SplitGS-Loc with those methods, we apply the feature rendering-based refinement of STDLoc [9], which renders feature and depth maps from the full GSFFs, and the RGB rendering-based refinement of GS-CPR [49], which establishes image correspondences between the rendered RGB image and the query using MASt3R [56]. As shown in Tab. A2, feature rendering-based refinement consistently improves SplitGS-Loc on both datasets. This improvement, however, requires storing the full GSFFs, drastically increasing the map size. GS-CPR further improves the 7Scenes result, while degrading performance on Cambridge Landmarks; we suspect that this difference stems from the fixed input resolution of MASt3R relative to the image resolution of each dataset. Overall, these results suggest that pose refinement can further improve SplitGS-Loc, but the gain comes with additional map storage and depends on the specific refinement method and its limitations.

Table A2: **Pose refinement results on Cambridge Landmarks [10] and 7Scenes [15].** We report the average of the per-scene median translation and rotation errors.

Method	Pose Refinement	Cambridge Landmarks		7Scenes	
		Size (MB)	Avg. (cm/°)	Size (MB)	Avg. (cm/°)
SplitGS-Loc (Ours)		97	8.7/0.15	88	1.0/0.26
+ Feature Rendering [9]	✓	415	**8.4/0.13**	888	0.9/0.23
+ GS-CPR [49]	✓	177	14.9/0.25	263	**0.6/0.19**

Table A3: **Localization results on Extended Cambridge Landmarks [57].** We report the average of the per-scene median translation and rotation errors.

Method	Mapping Time	Summer (cm/°)	Winter (cm/°)	Evening (cm/°)
STDLoc [9]	2h	**19.5**/0.34	20.1/0.36	21.1/0.37
SplitGS-Loc (Ours)	7min	**19.5/0.27**	**19.5/0.28**	**20.2/0.31**

F Robustness under Appearance Changes

We further evaluate SplitGS-Loc under appearance changes using Extended Cambridge Landmarks (ECL) [57]. ECL provides summer, winter, and evening variants of the Cambridge Landmarks test images while preserving their original camera poses. For both STDLoc [9] and SplitGS-Loc, we use the same maps as in the Cambridge Landmarks evaluation and directly evaluate them on ECL without adaptation. Tab. A3 shows that SplitGS-Loc consistently outperforms STDLoc across different appearance conditions. Since both methods use the same image feature extractor and original training views, the results highlight the robustness of our localization-specialized GSFFs construction to unseen appearance changes. Notably, SplitGS-Loc achieves this without adaptation to ECL, while retaining its efficient mapping process.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We explain the paper's contributions and scope accurately in the abstract and Sec. 1

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitation in Sec. 4.4.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We explain all the formulas in Sec. 3 and appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We explain the exact method and implementation details in Sec. 3 and Sec. 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our code is publicly available at `https://github.com/misoleekr/SplitGS-Loc`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all details in Sec. 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Following standard practice in this line of work, we do not report error bars. For fair comparison, our experiments follow the exact same configuration including random seeds as the baseline, and the train and test sets are completely separated.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We specify the compute resources in Sec. 4 and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We conducted the research with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: This is foundational research on camera pose estimation with no direct path to negative societal applications beyond those of existing methods in the field.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work uses public datasets and poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We checked the license of our baseline code and datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This work does not provide new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not contain crowdsourcing experiments and research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not need study participants.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: This work does not involve LLMs as a component of the core methods.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.