ULF-Loc: Unbiased Landmark Feature for Robust Visual Localization with 3D Gaussian Splatting

Supplementary Material

In this supplementary material, we first present a theoretical analysis of the inherent feature bias in Feature-3DGS [19], where Appendix A.1 establishes the problem setup and Appendix A.2 derives the bias expression under a simplified model. A more rigorous analysis under joint optimization is then provided in Appendix A.3, confirming the fundamental limitations of α-blending. Subsequently, Appendix B offers the pseudo-code for our key algorithmic components: Keypoint-Consensus Landmark Sampling (B.1) and the highly efficient, GPU-parallelizable Local Geometric Consistency Verification (B.2). Finally, a comprehensive set of experiments is detailed in Appendix C, encompassing the formal definitions of evaluation metrics (C.1), the use of semantic segmentation for enhancing 3DGS reconstruction (C.2), complete quantitative results on the 12Scenes (C.3) and Cambridge Landmarks (C.4) datasets, the localization speed comparison (C.5), ablation experiments on LGCV parameters (C.6), extensive qualitative visualizations (C.7), and a dedicated analysis of failure cases (C.8).

Contents

A. Theoretical Analysis of Feature Bias

A.1. Problem Setup and Notation

Consider a 3D feature point with feature vector denoted as $\mu \in \mathbb{R}^D$. Its 2D projection features across K training views are $\{f_k^{2D} \in \mathbb{R}^D\}_{k=1}^K$. We assume that the 2D feature f_k^{2D} in view k should preserve the characteristics of the true (but unknown) 3D feature μ, yet in practice exhibits variations due to viewpoint changes and other factors. We thus model this relationship as:

$$f_k^{2D} = \mu + \epsilon_k, \tag{1}$$

where ϵ_k represents view-independent variation, satisfying $\epsilon_k \sim \mathcal{N}(0, \Sigma)$ with $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_D^2)$. Our objective is to analyze the bias of features obtained by Feature-3DGS.

A.2. Bias Analysis under Simplified Model

In Feature-3DGS, the rendered feature $F_s(u_k)$ at pixel u_k in view k is computed via α-blending of multiple Gaussians:

$$F_s(u_k) = \sum_{i \in \mathcal{N}(u_k)} f_i \alpha_i T_i, \tag{2}$$

where $T_i = \prod_{j<i}(1-\alpha_j)$ is the accumulated transmittance, $\mathcal{N}(u_k)$ is the set of sorted Gaussians overlapping with pixel u_k, f_i is the feature vector of the i-th Gaussian, and α_i is its blending weight.

As described in the main paper, to analyze the inherent bias in the 3DGS optimization process, we decompose the feature rendering process. Assuming that the target Gaussian is at position t in $\mathcal{N}(u_k)$, we isolate its individual contribution:

$$F_s(u_k) = f_t \alpha_t T_t + \sum_{i \in \mathcal{N}(u_k), i \neq t} f_i \alpha_i T_i. \tag{3}$$

Defining the target's cumulative weight as $w_k = \alpha_t T_t$ and aggregating the remaining terms into a normalized neighborhood feature $B_k = (\sum_{i \in \mathcal{N}(u_k), i \neq t} f_i \alpha_i T_i)/(1 - w_k)$, we obtain the equivalent formulation:

$$F_s(u_k) = w_k f_t + (1 - w_k) B_k. \tag{4}$$

Feature-3DGS optimizes the features of each Gaussian primitive by minimizing the feature loss $\mathcal{L}_f$. Here, we employ the L_2 loss:

$$\mathcal{L}_f = \sum_{k=1}^{K} \| w_k f_t + (1 - w_k) B_k - f_k^{2D} \|_2^2. \tag{5}$$

To simplify and intuitively analyze the feature bias, we consider a randomly selected training view k and assume achieving perfect fitting in this view:

$$w_k f_t + (1 - w_k) B_k = f_k^{2D}. \tag{6}$$

Solving for the optimal feature f_t^* in this view yields:

$$f_t^* = \frac{f_k^{2D} - (1 - w_k)B_k}{w_k}, \quad w_k \neq 0. \tag{7}$$

Substituting $f_k^{2D} = \mu + \epsilon_k$:

$$f_t^* = \frac{\mu}{w_k} - \frac{(1 - w_k)B_k}{w_k} + \frac{\epsilon_k}{w_k}. \tag{8}$$

Since both w_k and B_k are random variables, we compute the expectation $\mathbb{E}[f_t^*]$ using conditional expectation ($\mathbb{E}[\mathbb{E}[X|Y, Z]] = \mathbb{E}[X]$). First, conditioning on w_k and B_k:

$$\mathbb{E}[f_t^*|w_k, B_k] = \frac{\mu}{w_k} - \frac{(1 - w_k)B_k}{w_k}, \tag{9}$$

Then, taking the overall expectation:

$$\mathbb{E}[f_t^*] = \mathbb{E}_{w_k, B_k}\left[\frac{\mu}{w_k} - \frac{(1 - w_k)B_k}{w_k}\right]. \tag{10}$$

The expected bias of this optimal solution relative to the true feature μ is:

$$bias = \mathbb{E}[f_t^*] - \mu = \mathbb{E}_{w_k, B_k}\left[\frac{1 - w_k}{w_k}(\mu - B_k)\right]. \tag{11}$$

This result aligns with Eq. (5) in the main paper, confirming the inherent bias in 3DGS feature learning. The bias expression reveals two critical insights:

- **Bias amplification from low contribution.** The factor $(1 - w_k)/w_k$ amplifies the bias when the target Gaussian's contribution to pixel rendering is partial ($w_k < 1$). This amplification becomes particularly severe when w_k approaches zero, corresponding to cases where the Gaussian is heavily occluded or has very low opacity.
- **Neighborhood feature entanglement.** The term $(\mu - B_k)$ quantifies the discrepancy between the true feature and the aggregated neighborhood features, meaning that each Gaussian's optimized feature deviates from its intrinsic characteristic μ to compensate for neighborhood context.

Notably, the bias vanishes only under two ideal conditions: (1) when $w_k = 1$ (complete dominance of the target Gaussian), or (2) when $B_k = \mu$ (perfect neighborhood consistency). In practice, occlusions, viewpoint variations, and scene complexity prevent these conditions from being met, making bias an inherent limitation of α-blending in 3DGS.

A.3. Rigorous Analysis under Joint Optimization

The previous simplified analysis provides an intuitive understanding of the source of bias. However, the actual optimization process in 3DGS involves joint optimization across all views. To establish a more rigorous theoretical foundation, we reanalyze the feature bias problem within the joint optimization framework.

Specifically, we obtain the optimal feature f_t^* by minimizing the loss function $\mathcal{L}_f$ (corresponding to Eq. (5)). According to the first-order necessary condition in optimization theory, the partial derivative of the loss function with respect to f_t should be zero at the extremum:

$$\frac{\partial \mathcal{L}_f}{\partial f_t} = 2\sum_{k=1}^{K} w_k \left(w_k f_t + (1 - w_k)B_k - f_k^{2D}\right) = 0. \tag{12}$$

Solving this yields the optimal feature f_t^*:

$$f_t^* = \frac{\sum_{k=1}^{K} w_k \left(f_k^{2D} - (1 - w_k)B_k\right)}{\sum_{k=1}^{K} w_k^2}, \quad \sum_{k=1}^{K} w_k^2 > 0. \tag{13}$$

Substituting $f_k^{2D} = \mu + \epsilon_k$:

$$f_t^* = \frac{\mu \sum_{k=1}^{K} w_k + \sum_{k=1}^{K} w_k \epsilon_k - \sum_{k=1}^{K} w_k(1 - w_k)B_k}{\sum_{k=1}^{K} w_k^2}. \tag{14}$$

Similarly, to compute the expectation $\mathbb{E}[f_t^*]$, we employ conditional expectation. First, conditioning on the sets $\{w_k\}_{k=1}^K$ and $\{B_k\}_{k=1}^K$:

$$\mathbb{E}[f_t^*|\{w_k\}, \{B_k\}] = \frac{\mu \sum_{k=1}^{K} w_k - \sum_{k=1}^{K} w_k(1 - w_k)B_k}{\sum_{k=1}^{K} w_k^2}. \tag{15}$$

Then, taking the overall expectation:

$$\mathbb{E}[f_t^*] = \mathbb{E}_{\{w_k\}, \{B_k\}}\left[\frac{\mu \sum_{k=1}^{K} w_k - \sum_{k=1}^{K} w_k(1 - w_k)B_k}{\sum_{k=1}^{K} w_k^2}\right]. \tag{16}$$

The expected bias of the optimal solution relative to the true feature μ is $bias = \mathbb{E}[f_t^*] - \mu$, namely:

$$bias = \mathbb{E}_{\{w_k\}, \{B_k\}}\left[\frac{\sum_{k=1}^{K} w_k(1 - w_k)(\mu - B_k)}{\sum_{k=1}^{K} w_k^2}\right]. \tag{17}$$

Compared to the simplified single-view analysis in Eq. (11), the rigorous joint optimization yields a more complex but structurally similar bias expression in Eq. (17), confirming that the feature bias is inherent to the α-blending process in 3DGS. This theoretical understanding motivates our proposed Geometry-Weighted Feature Fusion (GWFF) approach, which avoids α-blending optimization altogether and directly constructs unbiased features through multi-view aggregation.

B. Pseudo-code of K.C. Sampling and LGCV

B.1. Keypoint-Consensus Landmark Sampling

We present the pseudo-code of the Keypoint-Consensus Landmark Sampling (K.C. Sampling) algorithm in Algorithm 1. This algorithm implements an efficient sampling

Algorithm 1: Keypoint-Consensus Landmark Sampling

Require: Gaussians $\mathcal{G}$; Feature Extractor $\mathcal{F}$; Training Views $\mathcal{V}$; Distance Threshold τ_D; Landmark Number n; Nearest Neighbors k
Ensure: Sampled landmarks $\tilde{\mathcal{G}}$

$\mathcal{S} \leftarrow \mathbf{0}^{|\mathcal{G}|}$ % Consensus score per Gaussian
$\tilde{\mathcal{G}} \leftarrow \emptyset$
for $v \in \mathcal{V}$ **do**
  $\mathcal{P}_v \leftarrow \text{Project}(\mathcal{G}.\text{center}, v)$
  $\mathcal{I}_v \leftarrow \{i \mid \mathcal{P}_v^i \text{ within image bounds of } I_v\}$
  $\mathcal{K}_v \leftarrow \mathcal{F}(I_v)$ % Extract 2D Keypoints in view v
  for $i \in \mathcal{I}_v$ **do**
    $d_{\min} \leftarrow \min_{k \in \mathcal{K}_v} \|\mathcal{P}_v^i - k\|$
    if $d_{\min} \leq \tau_D$ **then**
      $\mathcal{S}[i] \leftarrow \mathcal{S}[i] + 1$
    end if
  end for
end for
$\mathcal{A} \leftarrow \text{RandomSample}(\mathcal{G}, n)$
for $a \in \mathcal{A}$ **do**
  $\mathcal{N}_a \leftarrow \text{kNN}(a, \mathcal{G}, k)$
  $g^* \leftarrow \arg\max_{g \in \mathcal{N}_a} \mathcal{S}[g]$
  $\tilde{\mathcal{G}} \leftarrow \tilde{\mathcal{G}} \cup \{g^*\}$
end for
return $\tilde{\mathcal{G}}$

strategy for selecting representative landmarks from Gaussian primitives through a two-stage process. First, it computes consensus scores by evaluating Gaussian projections against 2D keypoints across multiple views. Second, it applies random sampling with local KNN search to ensure geometric stability and uniform distribution.

B.2. Local Geometric Consistency Verification

We present the pseudo-code of the Local Geometric Consistency Verification (LGCV) algorithm in Algorithm 2. This algorithm is implemented in a tensor-operation style, which enables the full utilization of GPU parallel computing capabilities. Consequently, it achieves high computational efficiency and can rapidly process large-scale point set data.

C. More Experiments

C.1. Evaluation Metrics

We provide the formal definitions of the evaluation metrics used in the main paper. The rotation and translation errors between the estimated and ground-truth camera poses are calculated as follows:

$$\Delta\hat{\mathbf{R}} = \arccos\left(\frac{\text{trace}(\hat{\mathbf{R}}^\top \mathbf{R}) - 1}{2}\right), \tag{18}$$

$$\Delta\hat{\boldsymbol{t}} = \|\hat{\boldsymbol{t}} - \boldsymbol{t}\|_2, \tag{19}$$

Algorithm 2: Local Geometric Consistency Verification

Require: Source Points $X \in \mathbb{R}^{N\times 2}$; Target Points $Y \in \mathbb{R}^{N\times 2}$; Nearest Neighbors K; Angular Threshold τ_a; Scale Threshold τ_s; Support Score Threshold τ_{support}
Ensure: Valid match mask $M_{\text{valid}} \in \mathbb{B}^N$

$I \leftarrow \text{KNN}(X, K)$
% Gather neighbor coordinates: $N \times K \times 2$
$X_n \leftarrow \text{gather}(X, I)$, $Y_n \leftarrow \text{gather}(Y, I)$
% Compute relative vectors: $N \times K \times 2$
$\Delta X \leftarrow X_n - X[:, \text{None}, :]$, $\Delta Y \leftarrow Y_n - Y[:, \text{None}, :]$
% Normalize unit vectors
$\hat{X} \leftarrow \text{normalize}(\Delta X, \text{dim} = -1)$
$\hat{Y} \leftarrow \text{normalize}(\Delta Y, \text{dim} = -1)$
% Pairwise cosine similarities: $N \times K \times K$
$A_X \leftarrow \text{matmul}(\hat{X}, \hat{X}^\top)$, $A_Y \leftarrow \text{matmul}(\hat{Y}, \hat{Y}^\top)$
% Angular consistency mask
$M_{\text{angle}} \leftarrow |A_X - A_Y| < (1 - \tau_a)$
% Scale ratios: $N \times K$
$R \leftarrow \|\Delta Y\|_2 / (\|\Delta X\|_2 + \epsilon)$
% Expand dimensions
$R_k \leftarrow R[:, :, \text{None}]$, $R_j \leftarrow R[:, \text{None}, :]$
% $N \times K \times K \times 3$
$R_{\text{triplet}} \leftarrow \text{stack}([R_k, R_j, R_k \odot R_j, -1)$
% Scale consistency mask
$M_{\text{scale}} \leftarrow \text{var}(R_{\text{triplet}}, \text{dim} = -1) < \tau_s$
$S \leftarrow \text{sum}(M_{\text{angle}} \wedge M_{\text{scale}}, \text{dim} = (1, 2))$
return $M_{\text{valid}} \leftarrow S \geq \tau_{\text{support}}$

where $\hat{\mathbf{R}}$ and $\hat{\boldsymbol{t}}$ represent the estimated rotation matrix and translation vector, while $\mathbf{R}$ and $\boldsymbol{t}$ denote the corresponding ground-truth values.

C.2. Semantic Segmentation for Building 3DGS

Following the practice in [6, 9], we employ semantic segmentation during training on the Cambridge Landmarks dataset to enhance 3DGS reconstruction quality. We utilize Mask2Former [5] to precisely isolate dynamic objects (e.g., vehicles and pedestrians) and sky regions from the static background, as visualized in Fig. A. The segmentation masks are utilized during the 3DGS reconstruction process to exclude these transient elements, leading to cleaner and more consistent 3D representations. The removal of dynamic objects reduces the formation of ghosting artifacts, while the exclusion of sky regions eliminates unnecessary geometry in areas lacking structural information.

C.3. Complete Localization Results on 12Scenes

We present the complete localization results on 12Scenes in Tab. A, supplementing the four scenes shown in the main paper due to space constraints.

Table A. **Localization Results on 12Scenes.** We report the median translation and rotation errors (cm/°) for each scene. The last column reports the average over all 12 scenes.

	Methods	Apartment 1		Apartment 2				Office 1				Office 2		Avg.↓
		kitchen	living	bed	kitchen	living	luke	gates362	gates381	lounge	manolis	5a	5b	
APR	Marepo [4]	1.9/1.2	1.7/0.9	2.0/1.0	2.1/1.1	1.8/0.9	2.3/1.3	1.7/0.9	2.4/1.2	1.9/1.0	1.9/0.9	2.0/0.9	2.9/1.1	2.1/1.04
SCR	SCRNet [8]	2.3/1.3	2.4/0.8	3.3/1.5	2.1/1.0	4.2/1.8	4.4/1.4	2.6/0.8	3.4/1.4	2.7/0.9	1.8/1.0	3.6/1.5	3.4/1.2	3.0/1.22
	SCRNet-ID [10]	2.6/1.4	2.0/0.8	2.0/0.8	1.8/0.9	3.0/1.2	3.7/1.3	2.1/1.0	2.9/1.2	3.4/1.1	2.6/1.2	3.3/1.2	3.8/1.3	2.8/1.12
	DSAC* [1]	-	-	-	-	-	-	-	-	-	-	-	-	0.5/0.25
	ACE [2]	0.53/0.27	0.60/0.19	0.49/0.20	0.63/0.26	0.59/0.23	0.76/0.27	0.69/0.22	0.81/0.32	0.84/0.21	0.76/0.28	0.86/0.33	0.82/0.28	0.7/0.26
	GLACE [14]	0.49/0.27	0.59/0.19	0.49/0.22	0.58/0.27	0.60/0.24	0.69/0.31	0.67/0.22	0.72/0.28	0.67/0.20	0.65/0.28	0.79/0.31	0.80/0.25	0.7/0.25
	NeRF-SCR [3]	0.9/0.5	2.1/0.6	1.6/0.7	1.2/0.5	2.0/0.8	2.6/1.0	2.0/0.8	2.7/1.2	1.8/0.6	1.6/0.7	2.5/0.9	2.6/0.8	2.0/0.76
NeRF/GS	PNeRFLoc [17]	1.0/0.6	1.5/0.5	1.2/0.5	0.8/0.4	1.4/0.5	8.1/3.3	1.6/0.7	8.7/3.2	2.3/0.8	1.1/0.5	-	2.8/0.9	2.8/1.08
	SplatLoc [16]	0.8/0.4	1.1/0.4	1.2/0.5	1.0/0.5	1.2/0.5	1.5/0.6	1.1/0.5	1.2/0.5	1.6/0.5	1.1/0.5	1.4/0.6	1.5/0.5	1.2/0.50
	GSplatLoc [13]	1.31/0.24	0.68/0.21	1.42/0.24	0.67/0.26	0.54/0.19	1.84/0.31	1.82/0.26	1.99/0.26	0.64/0.22	0.78/0.27	3.66/0.37	0.94/0.27	1.4/0.26
	GSFFs-PR Feature [11]	0.3/0.20	0.3/0.18	0.4/0.17	0.7/0.42	0.4/0.21	0.6/0.27	0.5/0.23	0.5/0.27	0.8/0.29	0.5/0.22	0.9/0.41	1.1/0.41	0.6/0.27
	Marepo+GS-CPR [9]	0.67/0.35	0.40/0.19	0.41/0.21	0.45/0.25	0.59/0.25	1.04/0.45	0.86/0.35	0.52/0.26	0.76/0.20	0.48/0.22	0.69/0.27	0.99/0.28	0.7/0.28
	ACE+GS-CPR [9]	0.46/0.22	0.44/0.17	0.44/0.18	0.40/0.19	0.50/0.20	0.64/0.27	0.57/0.20	0.56/0.24	0.71/0.21	0.51/0.20	0.68/0.27	0.57/0.19	0.5/0.21
	STDLoc [6]	**0.26/0.17**	0.36/0.17	0.31/0.16	0.28/0.18	0.31/0.13	0.49/0.20	0.40/**0.14**	0.36/0.17	0.47/0.15	0.34/0.17	0.58/0.24	0.57/0.22	0.4/0.18
	Ours	0.29/0.19	**0.29/0.14**	**0.29/0.12**	**0.25/0.17**	**0.26/0.12**	**0.41/0.17**	**0.35/0.14**	**0.31/0.14**	**0.43/0.13**	**0.32/0.15**	**0.47/0.22**	**0.38/0.16**	**0.3/0.15**

Figure A. **Visualization of Segmentation Masks.** From left to right: query images, segmentation masks, and rendered images.

Table B. **Recall on Cambridge Landmarks Dataset.** We report the average recall (%) under different thresholds.

Methods	Cambridge Landmarks		
	Avg.↑[50cm/5°]	Avg.↑[15cm/5°]	Avg.↑[10cm/5°]
HLoc(SP+SG) [12]	91.4	64.8	52.0
ACE [2]	78.7	43.1	31.5
GLACE [14]	91.0	62.8	47.6
STDLoc [6]	**95.4**	70.8	59.9
GLACE+GS-CPR [9]	92.5	65.5	50.7
ACE+GS-CPR [9]	84.6	56.8	42.6
Ours	93.7	**72.0**	**62.2**

C.4. Additional Results on Cambridge Landmarks

Tab. B extends the Cambridge Landmarks evaluation from the main paper by reporting recall rates under the more stringent 10cm/5° threshold and adding comparisons with the structure-based method HLoc [12]. Our method achieves a recall of 62.2% at this challenging threshold, exceeding HLoc [12] by 10.2% and STDLoc [6] by 2.3%. These results demonstrate the particular advantage of our approach in high-precision localization.

C.5. Comparison of Localization Speed

We evaluate the computational efficiency of different methods by measuring their localization speed in frames per second (FPS). As illustrated in Fig. B, our method achieves a competitive speed of 4.4 FPS while maintaining superior localization accuracy. This represents a significant speed advantage over GSplatLoc [13] (0.6 FPS) and NeRFMatch [18] (2.2 FPS), while being comparable to STDLoc [6] (3.9 FPS). The speed-accuracy comparison reveals that our approach strikes an optimal balance, delivering both high precision and practical efficiency.

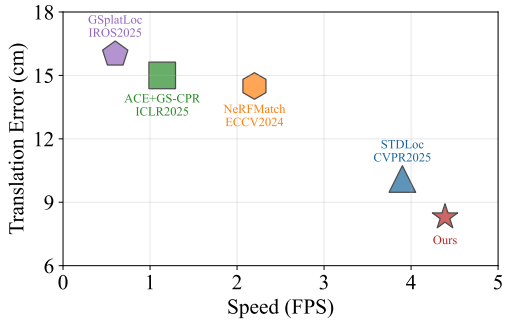

Figure B. **Average Median Translation Error and Speed Comparison on Cambridge Landmarks.**

C.6. Ablation Study on LGCV Parameters

We conduct the ablation study of LGCV parameters on the Cambridge Landmarks Court scene. We evaluate the performance by reporting the inlier rate (%) after mismatch

Table C. **LGCV Parameters Ablation on the Count Scene.** We report the inlier rate after removing mismatches with different parameter settings. The horizontal axis represents the scale threshold τ_s, and the vertical axis represents the angular threshold τ_a.

Inlier Rate (%)	$\tau_s = 0.05$	$\tau_s = 0.1$	$\tau_s = 0.2$	$\tau_s = 0.3$	$\tau_s = 0.4$
$\tau_a = 0.8059$	64.7	65.3	56.3	44.9	35.2
$\tau_a = 0.8559$	70.9	71.5	64.9	62.1	39.6
$\tau_a = 0.9059$	70.6	73.4	70.0	66.6	58.4
$\tau_a = 0.9659$	70.3	73.6	72.7	70.0	61.9
$\tau_a = 0.9759$	69.7	71.8	72.3	71.0	64.5
$\tau_a = 0.9859$	67.7	68.7	71.9	70.5	63.0

removal. The results under different parameter combinations of the angular threshold τ_a and the scale threshold τ_s are summarized in Tab. C. The experimental results reveal a clear trend: both overly lenient and overly strict threshold settings lead to a degradation in the inlier rate. Specifically, when the thresholds are too lenient (e.g., $\tau_a = 0.8059$, $\tau_s = 0.4$), the geometric constraints are insufficient to filter out all incorrect matches, resulting in a lower inlier rate. Conversely, when the thresholds are too strict (e.g., $\tau_a = 0.9859$, $\tau_s = 0.05$), the algorithm rejects a certain number of correct matches along with the outliers, thereby reducing the total number of valid correspondences and ultimately impairing the subsequent pose estimation. The ablation study identifies an optimal parameter range that balances the thoroughness of mismatch rejection with the preservation of correct matches. Based on these findings, we set $\tau_a = 0.9659$ and $\tau_s = 0.1$ for all experiments in the main paper.

C.7. Qualitative Visualizations

We present extensive qualitative evaluations on the 7Scenes, 12Scenes, and Cambridge Landmarks datasets in Fig. C. Each visualization is organized into four columns showing different stages of our localization pipeline. The first column displays the original query image. The second column visualizes 2D-3D correspondences by transforming them into 2D-2D matches. We render the sampled Gaussian landmarks after solving the initial pose to depict match quality. The third column presents the rendered image using this initial pose. The final column stitches the query image and the rendered image of the final pose to represent the localization result.

The visual results confirm the effectiveness of our K.C. sampling strategy and unbiased landmark features in generating reliable 2D-3D correspondences. Across all tested scenes, our method produces consistently high-quality matches that lead to accurate initial pose estimates. The rendered images in the third column closely match the corresponding query images, demonstrating the accuracy of these initial pose estimates. The precise alignment achieved in the final column further validates the high localization accuracy of our approach, demonstrating robust performance in both indoor and outdoor environments.

C.8. Failure Case Analysis

Fig. D shows some failure cases of our localization approach. The results indicate that our approach consistently achieves accurate initial pose estimation across these cases, demonstrating the robustness of the K.C. sampling strategy and the effectiveness of unbiased landmark features. These features, combined with high-quality Gaussian landmarks, enable reliable 2D-3D matching even in complex environments. However, the pose refinement stage occasionally fails due to limitations inherent in 3D Gaussian Splatting reconstruction [7, 15]. Specifically, we observe that floating artifacts near camera positions (a common issue in neural rendering techniques) produce erroneous renderings that mislead the image matching-based pose refinement stage, ultimately leading to incorrect pose estimation results. These failure cases indicate that employing a more robust refinement method could be a promising direction for future work.

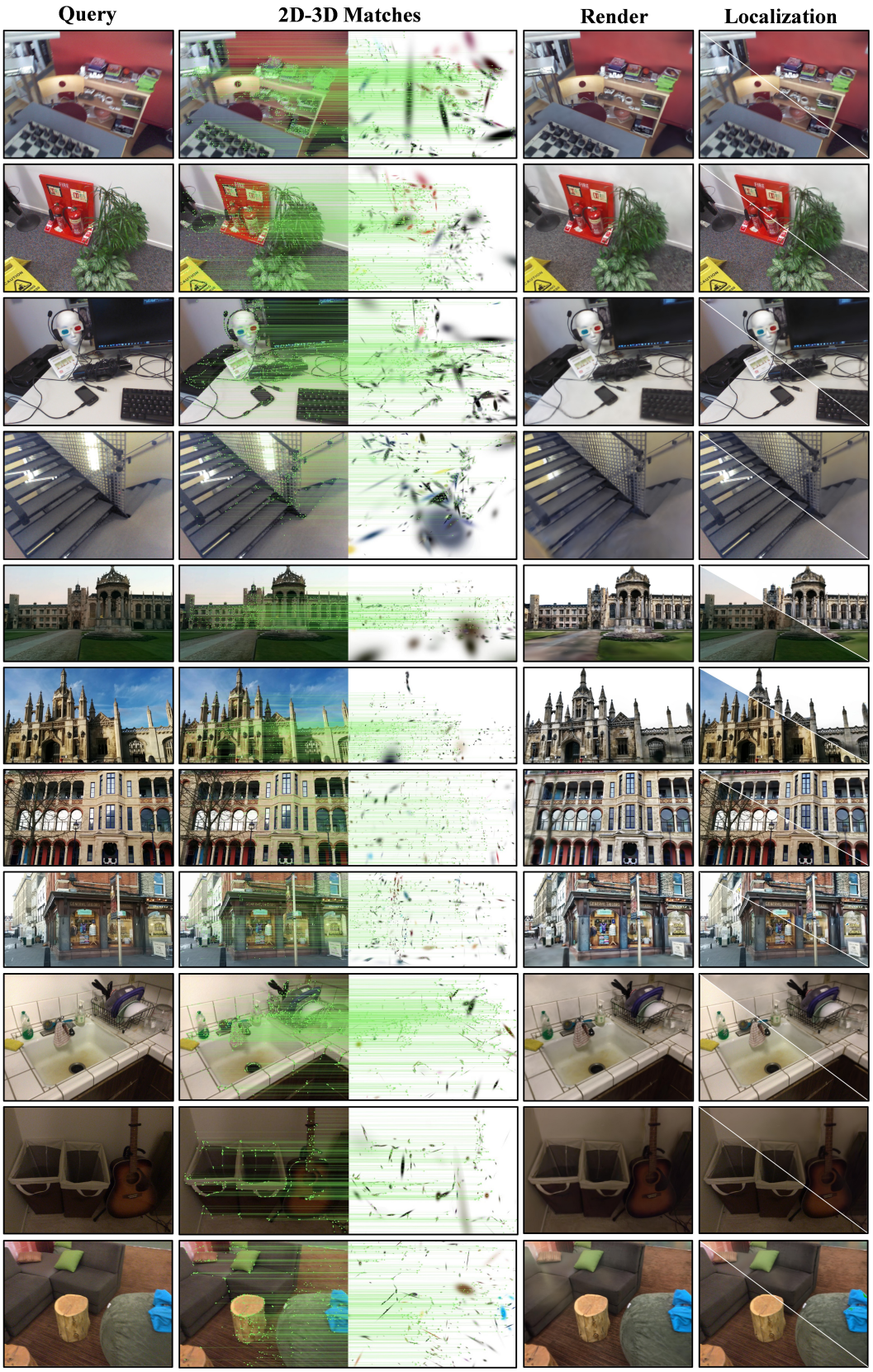

Figure C. **More Qualitative Visualizations on 7Scenes, 12Scenes, and Cambridge Landmarks.**

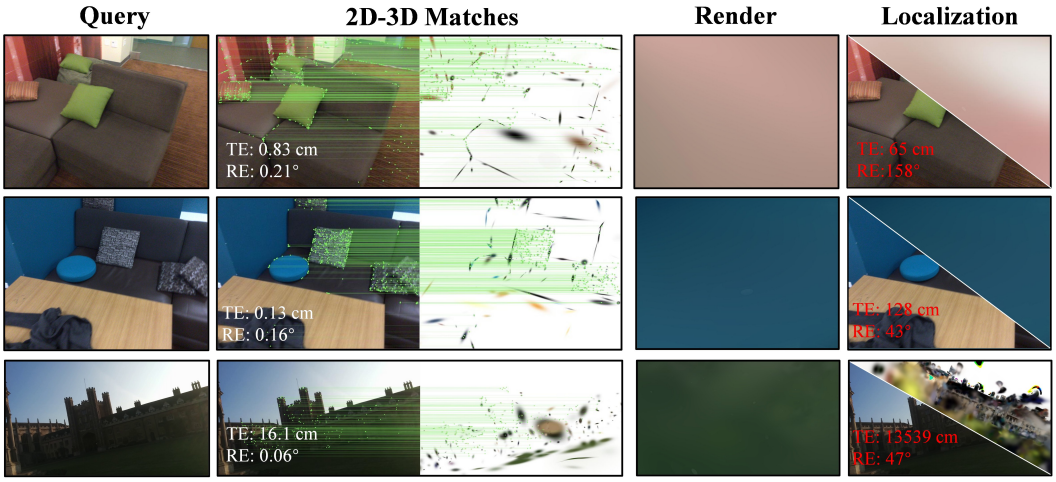

Figure D. **Failure Cases Visualization.** Initial pose estimation succeeds with robust 2D-3D matching, but the refinement stage fails due to floating artifacts.

References

[1] 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. 4

[2] 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. 4

[3] Le Chen, Weirong Chen, Rui Wang, and Marc Pollefeys. Leveraging neural radiance fields for uncertainty-aware visual localization. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6298–6305. IEEE, 2024. 4

[4] 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. 4

[5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 3

[6] 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. 3, 4

[7] 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. 5

[8] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11983–11992, 2020. 4

[9] 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. 3, 4

[10] Tony Ng, Adrian Lopez-Rodriguez, Vassileios Balntas, and Krystian Mikolajczyk. Reassessing the limitations of cnn methods for camera pose regression. In *International Conference on 3D Vision*, 2021. 4

[11] 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. 4

[12] 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. 4

[13] Gennady Sidorov, Malik Mohrat, Denis Gridusov, Ruslan Rakhimov, and Sergey Kolyubin. Gsplatloc: Grounding keypoint descriptors into 3d gaussian splatting for improved visual localization. *arXiv preprint arXiv:2409.16502*, 2024. 4

[14] 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. 4

[15] Yunsong Wang, Tianxin Huang, Hanlin Chen, and Gim Hee Lee. Freesplat++: Generalizable 3d gaussian splatting for efficient indoor scene reconstruction. *arXiv preprint arXiv:2503.22986*, 2025. 5

[16] Hongjia Zhai, Xiyu Zhang, Boming Zhao, Hai Li, Yijia He, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. Splatloc: 3d gaussian splatting-based visual localization for augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 2025. 4

[17] 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*, pages 7450–7459, 2024. 4

[18] 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. Springer, 2024. 4

[19] 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. 1