SplatLoc: 3D Gaussian Splatting-based Visual Localization for Augmented Reality

Hongjia Zhai[1], Xiyu Zhang[1], Boming Zhao[1], Hai Li[2], Yijia He[2], Zhaopeng Cui[1], Hujun Bao[1], and Guofeng Zhang[1]*

[1]State Key Lab of CAD&CG, Zhejiang University [2]RayNeo

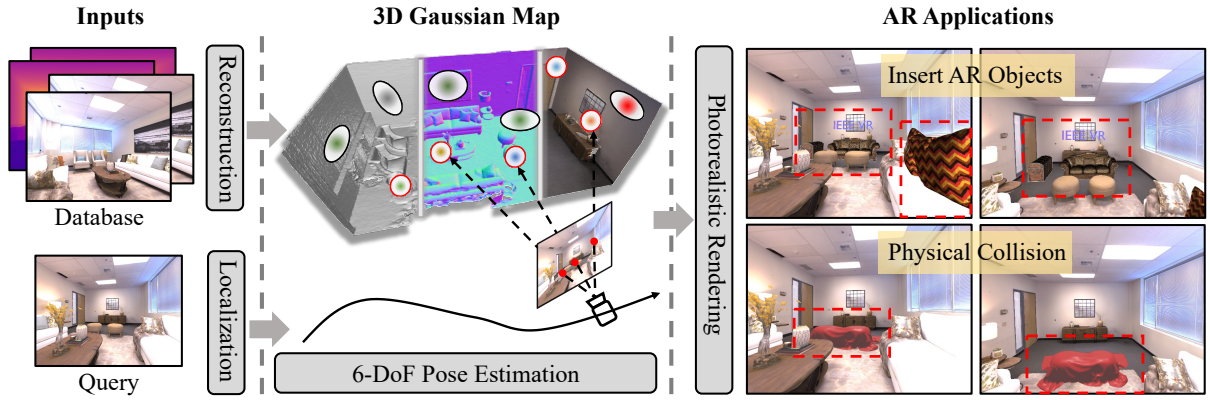


Figure 1: We present SplatLoc, an efficient and novel visual localization approach designed for Augmented Reality (AR). As illustrated in the figure, our system utilizes monocular RGB-D frames to reconstruct the scene using 3D Gaussian primitives. Additionally, with our learned unbiased 3D descriptor fields, we achieve 6-DoF camera pose estimation through precise 2D-3D feature matching. We demonstrate the potential AR applications of our system, such as virtual content insertion and physical collision simulation. We highlight virtual objects with red boxes. For AR demos, please refer to our supplementary video.

Abstract

Visual localization plays an important role in the applications of Augmented Reality (AR), which enable AR devices to obtain their 6-DoF pose in the pre-build map in order to render virtual content in real scenes. However, most existing approaches can not perform novel view rendering and require large storage capacities for maps. To overcome these limitations, we propose an efficient visual localization method capable of high-quality rendering with fewer parameters. Specifically, our approach leverages 3D Gaussian primitives as the scene representation. To ensure precise 2D-3D correspondences for pose estimation, we develop an unbiased 3D scene-specific descriptor decoder for Gaussian primitives, distilled from a constructed feature volume. Additionally, we introduce a salient 3D landmark selection algorithm that selects a suitable primitive subset based on the saliency score for localization. We further regularize key Gaussian primitives to prevent anisotropic effects, which also improves localization performance. Extensive experiments on two widely used datasets demonstrate that our method achieves superior or comparable rendering and localization performance to state-of-the-art implicit-based visual localization approaches. Project page: https://zju3dv.github.io/splatloc.

Index Terms: 3D Gaussian splatting, Visual localization, Novel view rendering, Implicit representation

1 Introduction

Visual localization is a critical technique that enables mobile devices or head-mounted displays to estimate the camera's 6-Degree-of-Freedom (6-DoF) pose relative to a pre-built 3D map. It plays an essential part in various Augmented Reality (AR) applications. For example, the visual localization approach can provide global 6-DoF pose information of the AR devices, which can be used to render virtual content in the real environment and facilitate the interaction of users with the physical space.

Generally, classical visual localization methods can be classified into two categories: regression-based and feature-based approaches. Regression-based approaches [3, 11, 17, 27, 41] usually use convolutional neural networks (CNN) to extract the high-level contextual feature of the image and encode the geometry information (*e.g.*, absolute pose, and scene coordinate) of the reconstructed environment. PoseNet [17] and SCRNet [27] are the representative works of directly regressing the pose or 3D coordinates of the pixels from the extracted feature of a single image. However, due to the lack of geometric constraints, these methods often lag behind feature-based approaches in terms of accuracy. Feature-based approaches [47, 61, 64] usually build a structure-based scene map beforehand (*e.g.*, 3D point cloud models) and associate each map primitive with one or more 3D descriptors. Those 3D consistent descriptors are usually obtained by performing multi-view fusion on the hand-crafted feature [31] or learning-based keypoint descriptors [9, 46] detected from 2D images. The detected 2D points in the query image can be matched against 3D descriptors to obtain 2D-3D correspondences for robust pose estimation [13, 22]. The localization performance of feature-based approaches is also decided by the repeatable and discriminative power of extracted descriptors. However, limited by the way of scene representation, these classical localization approaches can not perform photorealistic rendering, which is an essential part of AR applications.

In recent years, Neural Radiance Fileds (NeRF) [37] and 3D Gaussian Splatting (3DGS) [18] have emerged as the new paradigm for neural implicit scene representation. These paradigms use implicit representation (*e.g.*, multilayer perception [50], parametric

*Corresponding author: Guofeng Zhang.

encodings [25, 40, 62]) or explicit primitives (*e.g.*, points [56], 2D/3D Gaussians [14, 18]) to represent the scene property and achieve satisfactory performance of high-quality rendering and geometry reconstruction. Benefiting from differentiable NeRF-style volumetric rendering [16] and point-based alpha-blending [35], neural-based approaches can perform parameter optimization in an end-to-end way without 3D supervision. Inspired by [37, 56], some works [6, 29, 33, 60, 63, 64] use neural implicit representation to reconstruct the scene and perform pose estimation. iNeRF [29] is the first work that refines the 6-DoF pose via photometric error minimization between the query image and rendering results of the pretrained NeRF model. NeRF-SCR [6] and LENS [39] are the representative works that combine regression-based visual localization with neural radiance field. They train a scene-specific NeRF model to synthesize high-quality novel views to cover the whole scene space, providing additional training data for the optimization of their scene coordinate regress network. Similarly, the localization performance of these NeRF-aided regression approaches is also not competitive due to the lack of geometric constraints. To impose the geometric constraints, the feature-based method, PNeRFLoc [64], represents the scene with explicit structure [56] and associates each point in the map with learning-based descriptor [46]. Compared with [6, 39], PNeRFLoc [64] can achieve better localization performance and generalizability. However, PNeRFLoc, like traditional feature-based methods, requires explicit storage of point-wise features, which results in significant memory usage, making it impractical for mobile devices with limited storage.

To overcome the aforementioned limitations, we propose an efficient and novel visual localization method that achieves better performance with fewer model parameters, suitable for both localization and high-quality novel view rendering. Specifically, to reduce the model parameters, we do not store point-wise descriptors explicitly. Instead, we construct the feature volume from multi-view 2D feature maps and distill it to a scene-specific 3D feature decoder, which can avoid the descriptor bias of Gaussian primitives introduced by alpha-blending. We then propose an efficient salient 3D landmark selection algorithm to reduce the computational overhead of 2D-3D matching caused by a large number of Gaussian primitives. Finally, we perform position and scaling regularization for key Gaussian primitives to reduce the 3D center shift. Overall, the specific contributions of our proposed approach are summarized as follows:

- We propose an efficient and novel visual localization approach based on 3D Gaussian primitives that achieves accurate localization performance and high-quality, fast rendering with fewer parameters.
- We introduce an unbiased 3D descriptor learning strategy for precise 2D keypoints and 3D Gaussian primitives matching, using a scene-specific 3D feature decoder to regress the feature volume from multi-view feature maps.
- We develop an effective salient 3D landmark selection algorithm to reduce the number of primitives used for localization. Additionally, to mitigate the Gaussian primitive center shift induced by photometric rendering loss, we apply regularization on the location and scale of key Gaussian primitives.
- We conduct extensive experiments to demonstrate the state-of-the-art and comparable performance of visual localization and high-quality novel view rendering.

The rest of our paper is structured as follows: In Sec. 2, we provide a review of related works to contextualize our research. Next, in Sec. 3, we explain each key component in our reconstruction pipeline. Subsequently, in Sec. 4, we evaluate the performance of our system through various synthetic and real-world scenes. Finally, the conclusion and limitation are drawn in Sec. 5.

2 Related Works

In this section, we review the works most relevant to the proposed method, including visual localization, 3D Gaussian Splatting, and 3D feature field.

2.1 Visual Localization

Visual localization is one of the most commonly used localization techniques for applications [26, 38] of AR/VR, which can estimate the accurate 6-DoF pose of the camera. Generally, visual localization approaches can be divided into feature-based and regression-based methods. Regression-based approaches represent the map in an implicit way, which encodes the geometry information of the scene in the convolutional neural networks to regress the absolute pose [3, 17] and scene coordinates [11, 27]. However, both pose and coordinate regression methods need a large amount of 3D training data to make the scene-specific CNN model convergence and are not competitive in terms of localization accuracy. Feature-based approaches [24, 47, 61] obtain the pose via performing feature matching between the 2D keypoints of the query image and 3D points in the sparse Structure-from-Motion (SfM) model. According to the estimated 2D-3D correspondences, the pose of the query image can be computed via the Perspective-n-Point (PnP) algorithm [22]. The localization performance of feature-based methods relies on the discriminative ability of the hand-crafted [31] or learning-based keypoint descriptors [9, 46]. Recently, inspired by the neural implicit representation [37], some works have used neural implicit representation to aid visual localization. For example, [6] and [39] use a pretrained NeRF model to synthesize more high-quality novel views for the training of scene regression networks. PNeRFLoc [64] uses the explicit point-based neural representation to employ the geometry constraints and perform 2D-3D feature matching for 6-DoF pose estimation. However, similar to traditional feature-based methods, [64] also require a lot of memory to store the descriptors and matching graph for the sparse SfM model.

2.2 3D Gaussian Splatting

Recently, 3D Gaussian splatting (3DGS) [18] has shown great potential in 3D computer vision [5, 7, 14, 15, 28, 32, 54, 57]. Compared to the neural radiance field (NeRF) [37], 3DGS uses an explicit representation to reconstruct the scene and enable fast and high-quality view synthesis. Besides, 3DGS gets rid of the need for MLP and ray marching, which can accelerate the training speed for real-time rendering. Although 3DGS can achieve high-quality novel view rendering and fast training and rendering speed, they usually lead to poor geometric accuracy, *e.g*, the center of primitives are not accurately aligned with the surface. Some works aim to control the shape of the primitives [14], use unbiased depth rendering [5], and introduce geometric regularization during the optimization process, such as monocular depth [34, 57], and normals [52, 55]. Besides, some works [42, 43] pay attention to reducing the memory footprint of Gaussian splatting operations. Additionally, to enhance the scene understanding ability, [44, 48] assign an additional attribute, semantic feature, for each Gaussian primitive. To render the high-dimensional features, they perform quantization or dimensionality reduction, which leads to performance decreases for large indoor scenes.

2.3 3D Feature Field

Since the development of NeRF, some works have attempted to learn consistent 3D feature fields from the 2D feature map distilled from vision or language foundation models [4, 20, 45]. DecomposingNeRF [21] is the first work that adds an additional branch for the neural network to encode the semantic feature. The high-level contextual feature learned from [23, 4] can be used to distinguish the background and objects, which can be used for many editing tasks.

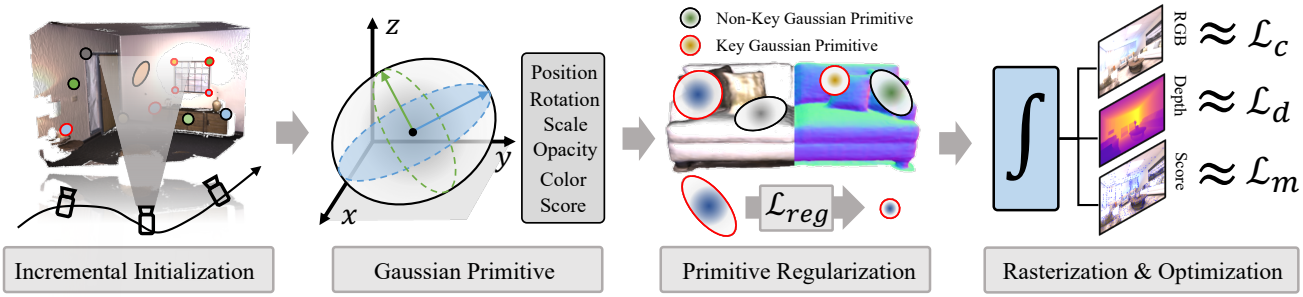


Figure 2: Reconstruction processes. We incrementally initialize the Gaussian primitives, and each primitive is associated with position μ, rotation q, scale s, opacity σ, color c, and 3D landmark score a. For key Gaussian primitives, we perform soft isotropy and scale regularization to mitigate the anisotropic results. The color loss $\mathcal{L}_c$, depth loss $\mathcal{L}_d$, 3D landmark loss $\mathcal{L}_m$, and regularization loss $\mathcal{L}_{reg}$ are used to optimize the properties of each primitive via differentiable rasterization.

To reduce the memory footprint, N3F [51] uses Principal Component Analysis (PCA) to compress the high dimensional DINO feature into lower dimensions for efficient 3D segmentation. Similar to NeRF-based methods, some recent works try to add feature attributes for Gaussian primitives. However, due to the memory demands of recording high-dimensional embeddings for each primitive, it is infeasible like the way in [21]. To tackle those problems, Shi *et al.* [48] propose an efficient quantization scheme to reduce the memory requirement of language embeddings. Qin *et al.* [44] train an auto-encoder to obtain a very low dimensional latent feature, which is associated with the Gaussian primitives. Those latent features can be decoded into high dimensions, which can be used for localization and segmentation. Learning feature fields via optimizing the similarity between rendered features and 2D feature maps from foundation models is not suitable for 2D-3D matching. There is a domain gap between the training and inference stages.

3 Method

In this section, we introduce the reconstruction and localization pipelines of our localization approach. In Sec. 3.1, we first introduce the 3D Gaussian scene representation. Then, we introduce the unbiased 3D descriptor feature field learning (Sec. 3.2). Besides, in Sec. 3.3, we introduce the 3D landmark selection strategy for localization and model compression. The regularization terms and overall optimization objectives of our reconstruction system are shown in Sec. 3.4 and Sec. 3.5. Finally, the visual localization process is introduced in Sec. 3.6.

3.1 3D Gaussian Scene Representation

In this part, we introduce the Gaussian primitive-based scene representation and incremental reconstruction process (Fig. 2).

Scene Representation. 3D Gaussian Splatting utilizes a collection of anisotropic 3D Gaussian primitives to represent the scene explicitly. Specifically, each Gaussian primitive is parameterized by its mean $\mu \in \mathbb{R}^3$ and covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$ defined in the world space, which is shown in the following:

$$G(\mu,\Sigma) = e^{-\frac{1}{2}(x-\mu)^T\Sigma(x-\mu)}. \tag{1}$$

To ensure the covariance matrix Σ maintains physical meaning during optimization, it is decomposed into a scaling matrix S and a rotation matrix R as proposed in [18]:

$$\Sigma = RSS^TR^T, \tag{2}$$

where the scaling matrix is computed from a 3D scale vector $\mathbf{s}$, $S = \mathrm{diag}([\mathbf{s}])$, and the rotation R is parameterized by quaternion.

Following the approach in Zwicker *et al.* [66], the 3D Guassians are projected into the 2D image plane for rendering. Given the viewing transformation matrix W and Jacobian J of the affine approximation of the projective transformation, the covariance matrix in the camera coordinates can be computed as:

$$\widetilde{\Sigma} = JW\Sigma W^TJ^T. \tag{3}$$

The corresponding 2D Gaussian distribution $\hat{G}(\widetilde{\mu},\widetilde{\Sigma})$ is derived from the 2D pixel location $\widetilde{\mu}$ of the 3D Gaussian primitive center and the projected covariance matrix $\widetilde{\Sigma}$.

Differentiable Rasterization. For novel view synthesis and fast rasterization-based rendering, each 3D Gaussian primitive is associated with an opacity $\sigma \in \mathbb{R}$ and a color $c \in \mathbb{R}^3$, represented using spherical harmonics (SH) coefficients. For photo-realistic rendering, the differentiable rasterizer adopts alpha-blending [35] to render the Gaussian property into the image plane, which accumulates Gaussian property and opacity values σ on a given pixel by traversing the ordered primitives. Specifically, the depth and color properties can be rendered with the following equation:

$$\hat{I} = \sum_{i=1}^{N} c_i \cdot \alpha_i \cdot \prod_{j=1}^{i-1}(1-\alpha_j), \quad \hat{D} = \sum_{i=1}^{N} d_i \cdot \alpha_i \cdot \prod_{j=1}^{i-1}(1-\alpha_j), \tag{4}$$

where $\hat{I}$, and $\hat{D}$ are the rendered color and depth, respectively. $\alpha_i = \hat{G}(\widetilde{\mu},\widetilde{\Sigma}) \cdot \sigma_i$ denotes the opacity contribution of each 2D pixel, $\prod_{j=1}^{i-1}(1-\alpha_j)$ is the accumulated transmittance, and N is the number of Gaussian primitives during the splatting process for a pixel.

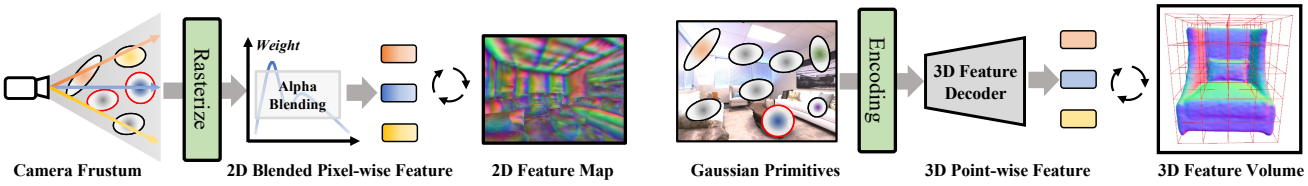
Additionally, to identify salient 3D landmarks for accurate and efficient visual localization, each Gaussian primitive is assigned a 3D landmark probability score $a \in \mathbb{R}$, representing the likelihood of the primitive being a key landmark in 3D space. Also, similar to the color and depth rendering equation, we can render the 3D landmark probability score into 2D with the following equation:

$$\hat{A} = \sum_{i=1}^{N} a_i \cdot \alpha_i \cdot \prod_{j=1}^{i-1}(1-\alpha_j). \tag{5}$$

Incremental Gaussian Initialization. During the reconstruction process, Gaussian primitives are initialized incrementally for each keyframe. For each incoming keyframe $\{I,D\}$, we first random sample pixels and project the sampled points into 3D space according to the camera pose, T_{wc}, and camera intrinsic, K, with the following equation:

$$\mu = T_{wc} \cdot \pi^{-1}(u, D(u)), \tag{6}$$

where $\pi^{-1}(\cdot)$ is the inverse of perspective projection, u is the sampled pixel. Then the Gaussian primitive is initialized with the projected 3D center μ and the color value from $I(u)$.



(a) Blending-based biased 3D feature learning. (b) Regression-based unbiased 3D feature learning.

Figure 3: Illustration of biased and unbiased 3D descriptor field learning. (a) The biased 3D feature optimization of previous works [44, 48], they use alpha-blending to obtain the 2D blended feature. (b) Our unbiased 3D feature learning scheme, which directly learns the 3D feature decoder from the constructed feature volume of multi-view feature maps.

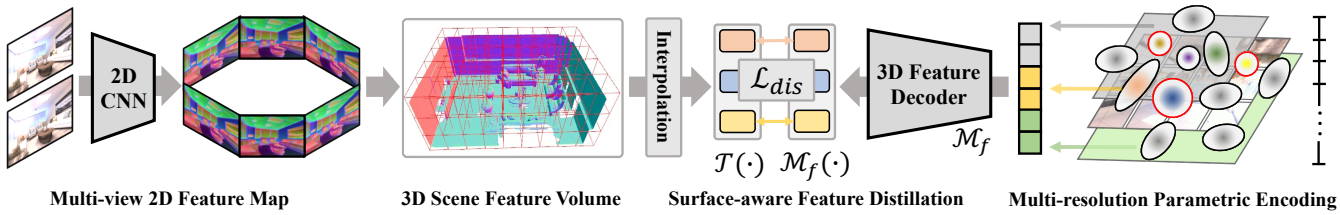


Figure 4: The pipeline of our unbiased 3D primitive descriptor learning. We first encode images based on the 2D CNN model [9] to obtain the multi-view feature maps and construct the 3D scene feature volume according to the depth and pose information. To enhance the representation ability of the 3D feature decoder, we use multi-resolution parametric encoding to aid the 3D scene-specific descriptor learning. Besides, we only sample descriptors on the scene surface for effective distillation.

Additionally,using the 2D keypoint score map generated by the SuperPoint model [9], we classify Gaussian primitives with scores exceeding a defined threshold as key Gaussian primitives, while others are marked as non-key primitives. Key Gaussian primitives typically exhibit higher 3D landmark probability scores and are more reliable for localization during the inference stage.

3.2 Unbiased 3D Descriptor Learning

To achieve effective 2D-3D feature matching while compressing the scene parameters, we propose learning descriptors for each Gaussian primitive using a 3D feature decoder. As shown in Fig. 3(a), previous approaches [44, 48] project 3D primitive feature into 2D blended feature via alpha-blending, then perform similarity optimization with CNN-generated 2D feature maps. However, such methods introduce a domain gap between the 2D CNN feature and the 3D Gaussian primitive descriptor, making them unsuitable for our task of 2D-3D feature matching. Furthermore, due to GPU memory constraints, directly learning high-dimensional descriptors is impractical. Existing solutions that rely on dimensionality reduction [44] or feature quantization [48] negatively affect localization performance. To address these issues, we propose an unbiased, accuracy-preserving 3D descriptor learning method (Fig. 3(b)), which directly regresses the 3D feature volume without any feature quantization and dimensionality reduction. The whole learning process is shown in Fig. 4.

2D Feature Map Lifting. To avoid the bias introduced by alpha-blending, we lift the 2D feature map into 3D feature volume and learn a 3D feature decoder to generate scene-specific descriptors for primitives. Specifically, for each keyframe, we first use a pre-trained CNN [9] to extract its 2D feature map, $f_i \in \mathbb{R}^{H\times W\times D_f}$, where D_f denotes the feature map dimension. According to the 6-DoF camera pose and intrinsics, we then project 2D feature maps into 3D feature volume, $\mathcal{V} \in \mathbb{R}^{\{D_x\times D_y\times D_z\times D_f\}}$, where D_x, D_y, and D_z are the volume dimensions based on scene size and voxel resolution. To model geometry information, we follow [8] and apply an additional TSDF volume to capture surface details from depth images. Besides, we update the 3D feature volume by fusing multi-view observations. Specifically, for each feature at position (x,y,z) in the volume, we apply weighted average pooling for the observation from different viewpoints:

$$\mathcal{V}[x,y,z] = \frac{w_i \cdot \mathcal{V}[x,y,z] + f_i(u)}{w_i + 1}, \tag{7}$$

where (x,y,z) is the volume coordinates derived from pixel location u and its corresponding depth value, and w_i is the weight of the current volume when fusing the i-th feature map f_i. So, we can obtain the feature of any spatial point in the scene through the trilinear interpolation operation.

Multi-resolution Parametric Encoding. As shown in [44, 50], a single MLP or encoder-decoder module leads to a performance decrease in scene representation and feature learning. To improve performance, we adopt multi-resolution hash feature encoding [40] to enhance the MLP decoder's ability to represent complex scenes, as shown on the right of Fig. 4. Specifically, for each Gaussian primitive with its 3D position μ we can obtain its multi-resolution parametric encoding $\mathcal{E}(\mu;\Theta)$ using the following equation:

$$\mathcal{E}(\mu;\Theta) = [\mathcal{T}(\mu,\theta_1),\cdots,\mathcal{T}(\mu,\theta_l)], \tag{8}$$

where Θ is the multi-resolution features $\Theta = \{\theta_l\}_{l=1}^{L}$, and $\mathcal{T}(\cdot)$ is the trilinear interpolation operation for each resolution level.

After obtaining the multi-resolution encoding feature, $\mathcal{E}(\mu;\Theta)$, the 3D descriptor of the primitive can be obtained with the 3D feature decoder, $\mathcal{D}$, which is implemented by a shadow MLP:

$$g = \mathcal{M}_f(\mathcal{E}(\mu;\Theta)), \tag{9}$$

where $g \in \mathbb{R}^{D_f}$ is the decoded high-dimensional descriptor at 3D position μ, which can be directly optimized with our constructed 3D feature volume $\mathcal{V}$.

Surface-aware Descriptor Distillation. Since much redundant information is stored in the feature volume $\mathcal{V}$ (*e.g.*, the descriptors in empty 3D space are often invalid and inaccurate), so we only perform descriptor distillation on the scene surface manifold. To achieve this, we first use the Marching Cubes algorithm [30] to find the surface inside the feature volume. Then, during the distillation

process, we randomly sample points on the surface and use the interpolation operation to obtain their corresponding features, which are then used to optimize the multi-resolution features Θ and the 3D feature decoder $\mathcal{M}_f$.

Algorithm 1: Salient 3D Landmarks Selection

Input: Primitives $\mathcal{P}$, Search radius r, Number N
Output: Selected landmarks $\mathcal{S}$

```
// Compute saliency score for all primitives
for p ∈ P do
    // Significance + Generalizability + Consistency
    H(p) = 2 · Sig(p) + min(2, Gen(p)) + Geo(p)
end
// Search N suitable landmarks
while |S| < N do
    // Saliency-based greedy search
    Ω = {p | p ∈ P, ||x − s|| > r, ∀s ∈ S};    // Candidates
    p = arg max H(p), for p ∈ Ω;
    if p exist then
        S = S ∪ p;    // Find the target landmark
    else
        r = r/2;    // Reduce search radius
    end
end
return S    // The final selected landmarks
```

3.3 Salient 3D Landmark Selection

Due to the need for photo-realistic rendering, 3DGS typically produces a large number of Gaussian primitives for indoor scenes (*e.g.*, around 400k points for `Room 0` in the Replica dataset). However, many of these primitives contain redundant information that is unnecessary for visual localization. For mobile or lightweight devices, using all these primitives would incur significant computational costs. To address this, we propose an efficient 3D landmark selection strategy based on the primitive saliency score, reducing the number of primitives required for localization.

Saliency Score Computation. Previous works [2, 10, 36, 59] have reduced the number of map points based on differentiable optimization or the importance of each point. However, there is no clear standard for evaluating the distinctiveness of points for accurate pose estimation. To reduce the number of primitives for localization, the 3D landmark selection method should find the suitable subset of Gaussian primitives from the reconstructed scene based on our defined saliency score. To select the informative 3D landmarks for localization, we define a saliency score for each Gaussian primitive $p \in \mathcal{P}$, which is computed based on the following criteria:

(1) Significance: Primitives with higher significance tend to be 3D landmarks, making them more useful for localization. As described in Sec. 3.1, each Gaussian primitive p is assigned a 3D landmark probability score a, representing the likelihood of being a landmark. So, for the significance term, we use the learnable landmark probability score a as the evaluation metric:

$$\mathrm{Sig}(p) = a. \tag{10}$$

(2) Generalizability: Primitives that were observed from many different viewing directions during the reconstruction are more generalizable and robust for localization. For this generalizability term, we use the largest angle between any two viewing directions as the evaluation metric which is defined in the following equation:

$$\mathrm{Gen}(p) = \max\{\arccos(\frac{(o_i-\mu)}{||(o_i-\mu)||}\cdot\frac{(o_j-\mu)}{||(o_j-\mu)||}) \mid o_i \in \mathcal{O}\}, \tag{11}$$

where o_i is the i-th camera center of database images, and μ is the location the primitive p.

(3) Geometry Consistency: The primitives with smaller multi-view geometric errors and aligned with the scene surface are more reliable for localization. To measure this term, we use the multi-view 3D distance error as the criteria, defined as:

$$\mathrm{Geo}(p) = \min(2, \frac{tr}{\mathrm{Mean}(\{dist_i\})}) + \min(2, \frac{tr}{\mathrm{Std}(\{dist_i\})}), \tag{12}$$

where tr is the distance error threshold parameter and $dist_i = \mu - d_i$ is the distance between the primitive position and its corresponding surface point observed by i-th camera.

Based on the above-mentioned criteria terms, we compute the final saliency score $\mathcal{H}(p)$ for all key Gaussian primitives:

$$\mathcal{H}(p) = 2\cdot\mathrm{Sig}(p) + \min(2, \mathrm{Gen}(p)) + \mathrm{Geo}(p), \tag{13}$$

where the constant coefficients and $\min(\cdot)$ are used for balance the weights for different terms.

3D Landmarks Selection. To select N effective and non-redundant 3D landmarks for visual localization, we apply a saliency-based greedy search algorithm, as illustrated in Algorithm 1. Our goal is to maximize the overall saliency score of selected landmarks and simultaneously ensure those landmarks cover the reconstructed scene evenly. This purpose is to enable the query cameras at different locations can observe the selected landmarks as much as possible.

Specifically, the landmark selection process starts by selecting the primitive with the highest saliency score as the initial landmark in the selected set $\mathcal{S}$. Next, we search for the primitive with the highest saliency score that is located at a distance greater than the search radius from any of the landmarks already in $\mathcal{S}$. If a suitable primitive is found, it is added to $\mathcal{S}$. If no suitable primitive is found, the search radius is reduced until a suitable primitive is identified. The process continues until the size of the selected set $\mathcal{S}$ reaches the target number N, at which point the greedy search stops.

3.4 Key Gaussian Primitive Regularization

In the incremental initialization of Gaussian primitives (Sec. 3.1), the primitives initialized by the 2D keypoints are designated as key Gaussian primtives. However, due to the lack of multi-view geometry optimization, the centers of those key Gaussian primitives are not accurately aligned with the surface and can shift away from their original 3D centers. This drift occurs as the primitives adjust to better appearance fitting. Such geometry errors can result in inaccurate 3D landmark learning since the misaligned primitives can introduce errors in localization. To address this, we apply position and scale regularization to the key Gaussian primitives, as illustrated in the middle of Fig. 2). During the differentiable optimization process, we prevent the centers of key Gaussian primitives from being updated to preserve their original alignment. Additionally, we use the 2D keypoint score map generated by [9] to guide a soft isotropy and scale regularization, which penalizes the scaling parameters as follows:

$$\mathcal{L}_{reg} = \sum_{\mathbf{s}, m\in\mathcal{P}_k} |\mathbf{s} - \delta\cdot(1-m)|, \tag{14}$$

where $\mathbf{s}$ and m are the 3D scale vector of the Gaussian primitive and its 2D keypoint score, $\mathcal{P}_k$ is the set of key Gaussian primitive set, and δ is the scale threshold for key Gaussian primitive.

This regularization term has two purposes. The first one is to control the 3D scale of the 3D key Gaussian primitives based on its 2D keypoint score reducing their influence during rasterization. The other one is to ensure that key primitives remain isotropic by enforcing a uniform scale constraint across all three axes.

3.5 Objective Functions

We adopt five different objective functions to optimize our system.
Reconstruction Loss. Similar to the [18, 34], for each pixel with ground truth depth and color, we apply the reconstruction losses between the rendered value and ground truth value measured by the camera. The color and depth reconstruction loss is shown in the following:

$$\mathcal{L}_c = \sum_{i=1} |\hat{I}(i) - I(i)|, \quad \mathcal{L}_d = \sum_{i=1} |\hat{D}(i) - D(i)|, \tag{15}$$

where $\hat{I}$ and $\hat{D}$ are the rendered color and depth, which are computed by Eq. (4).
3D Landmark Loss. As shown in Sec. 3.1, each Gaussian primitive is associated with a 3D landmark probability score, a. Due to the lack of 3D information, we can not directly optimize the 3D landmark probability score of the primitive. So, to optimize the probability score, we apply loss between the rendered 2D probability score map $\hat{A}$ (Eq. (5)) and the 2D probability score map A generated by the [9]. The loss is calculated on all rendered pixels:

$$\mathcal{L}_m = \sum_{i=1} \texttt{BCE}(\hat{A}(i), A(i)), \tag{16}$$

where $\texttt{BCE}(\cdot)$ is the binary cross-entropy loss.
Descriptor Distillation Loss. We use the surface-aware descriptor distillation loss term to optimize the scene-specific multi-resolution feature and 3D feature decoder, which is defined as follows:

$$\mathcal{L}_{dis} = \sum_{x \in \mathcal{X}} |1 - \cos(\mathcal{M}_f(\mathcal{E}(x;\Theta)), \mathcal{T}(x;\mathcal{V}))|, \tag{17}$$

where $\mathcal{X}$ is the sampled point set on the scene surface manifold, and $\cos(\cdot)$ denotes the cosine similarity of two descriptors.
Regularization Loss. To mitigate the anisotropic (arrow-shaped) Gaussian primitives and limit the influence of key Gaussian primitives, we adopt the scale and isotropy regularization loss term $\mathcal{L}_{reg}$ (Eq. (14)) introduced in Sec. 3.4.

So, the final loss function is presented as follows:

$$\mathcal{L} = \lambda_1 \cdot \mathcal{L}_c + \lambda_2 \cdot \mathcal{L}_d + \lambda_3 \cdot \mathcal{L}_m + \lambda_4 \cdot \mathcal{L}_{dis} + \lambda_5 \cdot \mathcal{L}_{reg}, \tag{18}$$

where $\{\lambda_i\}$ are the weights for each optimization component.

3.6 Localization

Once we obtain the reconstructed 3D Gaussian primitives, we can estimate the 6-DoF pose of the query image during the inference stage. For the query image, I_q, we first extract the 2D keypoints with descriptors $\{f_i\}$ via the pre-trained 2D CNN model, SuperPoint [9]. For the 3D implicit map, we use all key Gaussian primitives or selected 3D landmarks inside the frustum of the reference image for feature matching. The reference image in the reconstruction database is obtained via image retrieval with the image-level descriptor [1], and the 3D descriptors $\{g_i\}$ of the selected primitives are computed via the scene-specific 3D feature decoder and multi-resolution parametric encoding Eq. (9). To estimate the 2D-3D correspondence, we used the cosine similarity between the 2D and 3D descriptors, $\{f_i\}$ and $\{g_i\}$, as the measure criteria. So, we can use the RANSAC+PnP [12, 22] algorithms to estimate the 6-DoF pose based on the estimated correspondences.

4 Experiments

In this section, we first introduce the used datasets and provide the implementation details of our approach. Then, we evaluate our visual localization and rendering performance. When compared with multiple baselines, we highlight the best two results with different colors.

4.1 Datasets

We evaluate the performance of our method on a variety of scenes from two commonly used datasets. The Replica [49] and 12-Scenes [53] both contain high-quality RGB-D sequences of various indoor scenes. For the Replica dataset, we take 8 synthetic scenes provided by [65]. Each scene contains two sequences, and each sequence includes 900 RGB-D frames. Following [6], we use the first sequence as the training set, and the second sequence is used for evaluation. For the 12-Scenes dataset, each scene contains different numbers of RGB-D sequences. We follow the common setting [6, 53], the first sequence is used for evaluation and others are used for the training set.

4.2 Implementation Details

For the incremental Gaussian initialization process, we project all pixels with 2D keypoints score larger than a threshold 0.005 into 3D space for key Gaussian primitive initialization. For the remaining pixels, we random sample $1/64$ points from each keyframe for initialization. The 3D feature decoder is a 4-layer MLP with 128 hidden units, and the final output dimension of the descriptor is set to 256. The finest resolution of multi-resolution parametric encoding is 6cm. The learning rates of position, color, opacity, 3D landmark probability, scale vector, and rotation for each primitive are set to 1.6e-4, 2.5e-3, 0.05, 0.05, 0.001, and 0.001. Besides, the learning rates of encoding parameters and 3D feature decoder are both set to 1e-3. We use Adam optimizer for the optimization of Gaussian primitives, multi-resolution encoding parameters, and 3D feature decoder. For the weight of objective functions in Eq. (18), we set $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5\}$ to 1.0, 0.5, 1.0, 1.0, and 0.01, respectively. After each incremental Gaussian initialization, we random sample 5 frames to construct the keyframes for optimization with 10 iterations. When we finish the primitive initialization and geometry optimization, we perform additional appearance refinement with 30k iterations. We adopt the same density strategy in the original 3DGS [18].

4.3 Evaluation of Localization

In this part, we show the localization performance on Replica [49] and 12-Scenes [53] datasets. Following the common setting [6], we take the images in the second sequence of Replica and the first sequence of 12-Scenes as the test set. The compared baseline approaches, evaluation metrics, and results of this part are shown in the following.
Baseline and Metrics. For the evaluation of localization, we follow the setting of [6] and take five approaches as our compared baselines, including four regression-based methods: SCRNet [27], SCRNet-ID [41], SRC [11], NeRF-SCR [6], and one recent feature-based method PNeRFLoc [64]. The values of PNeRFLoc [64] are the PnP results obtained with their released codes, and the values of other compared methods are taken from the [6]. To measure the localization accuracy of the different approaches, we use the commonly used relative translation error Eq. (19) and relative rotation error Eq. (20) as the evaluation metrics:

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

$$\Delta R = \arccos((\text{Tr}(R^T \hat{R}) - 1)/2), \tag{20}$$

where t and R are the ground-truth translation and rotation, respectively, and $\hat{t}$ and $\hat{R}$ are the estimated ones. And $\text{Tr}(\cdot)$ is the trace of the matrix.
Qualitative and Quantitative Results. To validate the effectiveness of our method, we perform per-frame pose estimation for the query images inside the test set. The quantitative localization results of Replica [49] and 12-Scenes [53] datasets are shown in Tab. 1 and Tab. 2, respectively. As can be seen from Tab. 1, compared with the previous best baseline, PNeRFLoc [64], we achieved

Table 1: Visual localization performance on Replica dataset. We report median translation and rotation errors (cm, degree).

Method	Room 0	Room 1	Room 2	Office 0	Office 1	Office 2	Office 3	Office 4
SCRNet [27]	2.05 / 0.33	1.84 / 0.34	1.31 / 0.26	1.69 / 0.34	2.10 / 0.52	2.21 / 0.41	2.13 / 0.37	2.25 / 0.43
SCRNet-ID [41]	2.33 / 0.28	1.83 / 0.35	1.78 / 0.29	1.79 / 0.37	1.65 / 0.42	2.07 / 0.37	1.79 / 0.28	2.42 / 0.35
SRC [11]	2.78 / 0.54	1.92 / 0.35	2.97 / 0.63	1.45 / 0.30	2.07 / 0.53	2.53 / 0.51	3.44 / 0.63	4.84 / 0.90
NeRF-SCR [6]	1.53 / 0.24	1.96 / 0.31	1.34 / 0.22	1.61 / 0.35	1.54 / 0.44	1.69 / 0.33	2.40 / 0.38	1.69 / 0.32
PNeRFLoc [64]	1.00 / 0.21	1.32 / 0.28	1.43 / 0.29	0.72 / 0.15	1.08 / 0.28	1.71 / 0.37	2.39 / 0.30	1.63 / 0.32
Ours	**0.53 / 0.10**	**1.06 / 0.20**	**1.05 / 0.22**	**0.42 / 0.08**	**0.85 / 0.21**	**1.25 / 0.24**	**1.30 / 0.22**	**1.10 / 0.19**

Table 2: Visual localization performance on 12-Scenes dataset. We report median translation and rotation errors (cm, degree).

Scenes	Apartment 1		Apartment 2				Office 1				Office 2	
Method	kitchen	living	bed	kitchen	living	luke	gates362	gates381	lounge	manolis	5a	5b
SCRNet [27]	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
SCRNet-ID [41]	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
NeRF-SCR [6]	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
PNeRFLoc [64]	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	X	2.8 / 0.9
Ours	**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**

Table 3: Novel view synthesis performance on Replica dataset. All scenes are evaluated on the novel views in the test sequence. We outperform PNeRFLoc [64] on the commonly used rendering metrics on all scenes.

Method	Metric	Room 0	Room 1	Room 2	Office 0	Office 1	Office 2	Office 3	Office 4	Avg.
PNeRFLoc [64]	**PSNR** ↑	26.67	24.03	27.43	31.44	26.43	25.78	23.33	21.62	25.84
	SSIM ↑	0.8730	0.8387	0.8823	0.9355	0.8874	0.8617	0.8026	0.7310	0.8515
	LPIPS ↓	0.1228	0.2231	0.2603	0.1026	0.1397	0.1704	0.4078	0.5259	0.2440
Ours	**PSNR** ↑	27.55	27.48	30.55	35.756	33.62	27.34	29.86	28.99	30.14
	SSIM ↑	0.8873	0.8885	0.9396	0.9674	0.9508	0.9158	0.9407	0.9171	0.9259
	LPIPS ↓	0.1034	0.1106	0.0719	0.0345	0.0624	0.0972	0.0659	0.1024	0.0810

the best localization results on all scenes in the Replica dataset with the lowest translation and rotation errors.

For the localization results in the challenging 12-Scenes dataset [53], the performance of the point-based neural localization approach, PNeRFLoc [64], is not as good as in the Replica dataset. Compared with the NeRF-based scene coordinate regression approach, NeRF-SCR [6], PNeRFLoc shows worse performance on two scenes with translation errors larger than 5cm and rotation errors larger than 3 degrees (scene `luke` and `gates381`). Besides, PNeRFLoc totally failed on scene `5a`, which is denoted as 'X'. Compared with those baselines, our approach has achieved excellent localization results, achieving the best localization performance in 11 scenes. Except for the scene `Apt2/kitchen`, our performance is slightly lower than [64]. Among these 12 scenes, our approach can achieve stable localization results. The reported median errors for all scenes, the translation, and the rotation of our method are less than 1.6 cm and 0.6 degrees, respectively. Our approach can achieve more competitive performance localization results among different scenes than the baselines.

4.4 Evaluation of Rendering

In this part, we evaluate the performance of novel view rendering on Replica [49] dataset. Following the common setting [6], we take the images in the second sequence as the test set. The compared baseline methods, evaluation metrics, and results of this part are shown below.

Baseline and Metrics. For the evaluation of novel view rendering, we take the recent neural-based visual localization approach, PNeRFLoc [64], as the compared baseline. The values of PNeRFLoc [64] are obtained with their released codes. To evaluate the rendering performance, we adopt the commonly used metrics, Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS).

Qualitative and Quantitative Results. We show the novel view rendering performance on each viewpoint in the test set. The quantitative results of Replica [49] dataset are reported in Tab. 3. The results show that our approach significantly outperforms the compared baseline on all scenes by a substantial margin. Compared with PNeRFLoc [64], we achieve the 4.302, 0.074, and 0.163 performance improvement on PSNR, SSIM, and LPIPS metrics, respectively. Besides, we also show the qualitative novel view rendering results of several selected scenes in Fig. 5. From the figure, we can see that our method can generate clearer results than PNeRFLoc [64]. PNeRFLoc cannot model the high-frequency appearance details of the scene very well and may produce artifacts and blurry rendering results from novel viewpoints. In contrast, our method can better reconstruct the appearance information of the scene and have better rendering results from novel viewpoints. As shown in the Tab. 3, higher LPIPS indicates higher perceptual similarity of images rendered by our method.

Table 4: Training time, memory usage, and rendering FPS of different methods on scene `manolis` from 12-Scenes Dataset.

Method	Training Time↓	Memory↓	FPS↑
SCRNet [27]	2 days	165 MB	-
NeRF-SCR [6]	16 hours	-	-
PNeRFLoc [64]	1 hour	788 MB	0.23
Ours	25 mins	112 MB	498

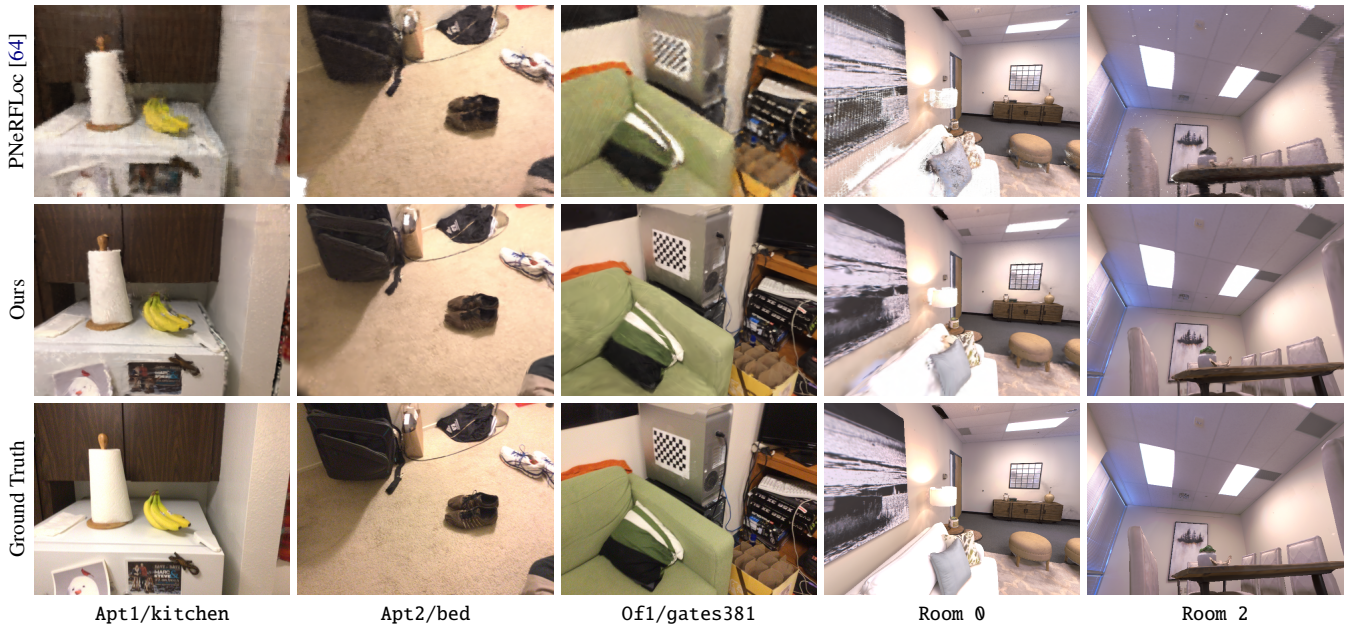


Figure 5: Visualization of novel view synthesis. We show some novel view rendering results from different scenes. From top to bottom, there are results of PNeRFLoc [64], ours, and ground truth. Our rendering results are more clear and have less noise information.

4.5 Memory Usage and Time Analysis

In this part, we measure the memory usage, method training time, and rendering speed of the neural-based localization approach. The results are shown in Tab. 4, evaluated in the scene `manolis`. Due to SCRNet [27] can not perform novel view rendering, we do not report it rendering FPS performance. And NeRF-SCR [6] don't release their codes and do not report their memory usage, FPS in their paper. So, we also don't report them in the table. As can be seen from the table, our method has a faster training speed, smaller model storage, and faster rendering speed. Scene coordinate regression-based approaches, *e.g.*, SCRNet [27] and NeRF-SCR [6], need more than ten hours or even days for the convergence of the regression network. Due to the fast differentiable rasterization process, our approach has a faster convergent speed than the point-based neural implicit approach, PNeRFLoc [64] (25 mins V.S. 1 hour). Due to the fact that we learn the descriptor from the 3D feature decoder not the storage descriptor for each primitive, our storage memory is 7x less than [64] (112 MB V.S. 788 MB). Besides, for the rendering speed (FPS) at the resolution of 640×480, our rendering speed is 2k times faster than [64] (498 FPS V.S 0.23 FPS). Our approach can not only estimate the accurate 6-DoF pose for the query camera but also perform real-time high-quality novel view rendering performance, which is more suitable for virtual reality and augmented reality.

4.6 Ablation Studies

In this part, we conduct ablation studies to validate the effect of each part or design in our system.

Effects of Different Descriptor Learning Settings. In Tab. 5, we show the performance of using different kinds of approaches to learning descriptors for Gaussian primitive on two selected scenes. #1 represents the way used in [44], which uses an auto encoder-decoder (Auto E.D.) to compress the high dimensional feature and perform biased alpha-blending. #2 and #3 represent that using our 3D feature decoder without/with positional encoding, respectively. #4 represents our approach with our muti-resolution parametric encoding but without surface-aware descriptor distillation (S.D.D). We can draw some conclusions from the results in the table. Comparing #2, #3, and #4, the descriptor obtained by just optimizing a 3D feature decoder is not accurate enough. In large indoor scenes, the representation ability of an MLP is not enough to learn scene features with sufficiently high discrimination ability, and additional parametric encoding needs to be used to assist the descriptor learning of 3D scenes. Comparing #1 and #4, our feature learning method is more accurate than the alpha-blending method. Comparing #4 and #5, we show that paying more attention to the 3D points on the scene geometry surface can lead to better and faster convergence for primitive descriptor learning. Compared with the baselines, our unbiased 3D descriptor learning approach can achieve better localization results.

Table 5: Visual localization performance with different descriptor learning settings. We report the median translation and rotation errors (cm, degree) on two selected scenes, `Room 0` from Replica and `apt1/kitchen` from 12-Scenes.

Case	Description	Room 0	Apt1/Kitchen
#1	Blending w/ Auto E.D.	2.97 / 1.69	3.90 / 2.69
#2	Ours 3D Dec. w/o P.E.	82.97 / 16.68	13.27 / 8.13
#3	Ours 3D Dec. w/ P.E.	6.87 / 4.60	7.53 / 3.71
#4	Ours w/o S.D.D	1.14 / 0.58	1.86 / 1.02
#5	Ours Full	**0.53 / 0.10**	**0.77 / 0.43**

Different Resolution of Parametric Encoding. In Sec. 3.2, we adopt multi-resolution parameter encoding to enhance the representation ability of the single MLP decoder. We investigate the performance of the spatial resolution used for encoding feature grids. The results on two selected scenes (`Apt1/kitchen` from 12-Scenes and `Office 1` from Replica) are shown in Fig. 6. As shown in the figure, when the resolution of parametric encoding is greater than 20 cm, the localization accuracy will quickly decrease as the resolution increases. When the resolution is below 10 cm, the localization accuracy will converge to a stable value. So, we set the resolution as 6 cm for all scenes in our experiments.

Effects of Different Numbers of 3D Landmarks. We show the localization performance of using the different numbers of selected

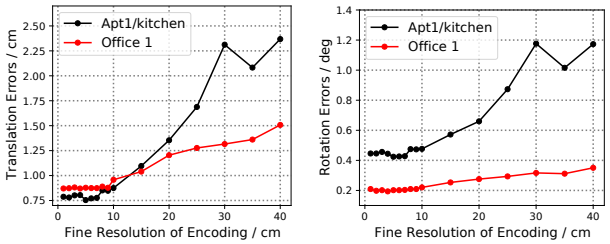


Figure 6: Visual localization performance of using different resolutions of parametric encodings. We report median translation and rotation errors (cm, degree) on two selected scenes.

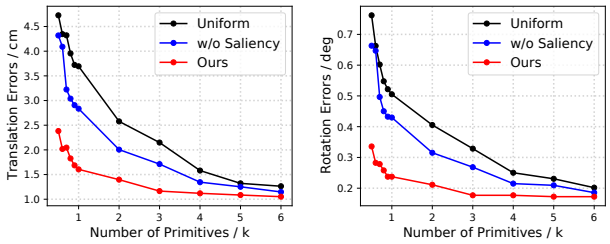


Figure 7: Visual localization performance of using different numbers of 3D landmark. We report median translation and rotation errors (cm, degree) on the scene of `Room 0` from the Replica Dataset.

landmarks. To validate the effectiveness of our proposed selection algorithm (Sec. 3.3), we take several comparison baselines into consideration. The localization results are shown in Fig. 7. 'Uniform' denotes the performance uniform downsample of the Gaussian primitives. 'w/o saliency' denotes the performance achieved by the greedy search method but without our saliency score for optimal selection. As can be seen from the figure, our proposed landmark selection algorithm can lead to better performance when only a small number of 3D landmarks are used for localization ($\sim$ 500 landmarks). Those results without using greedy search and saliency score perform worse results for the situation of small primitives. Compared with the baselines, our 3D landmark selection algorithm can achieve better localization results.

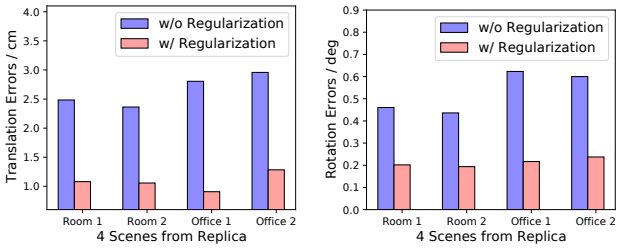


Figure 8: Visual localization performance of using primitive regularization terms. We report median translation and rotation errors (cm, degree) on four selected scenes from Replica.

Effectiveness of Key Gaussian Primitive Regularization. We validate the performance of key Gaussia primitive regularization (Sec. 3.4) used for our visual localization approach on four selected scenes in the Replica dataset [49]. The localization results are shown in Fig. 8. 'w/o Regularization' and 'w/ Regularization' represent the performance of not using and using position and scale regularization terms for key Gaussian primitives, respectively. Using regularization terms can lead to more consistent and accurate 3D landmark learning. As can be seen from the figure, with regularization terms, we can achieve better and more stable localization performance improvement. Across the selected four scenes, using regularization terms can reduce the error by an average of 1.57 cm and 0.32 degrees.

4.7 AR Applications

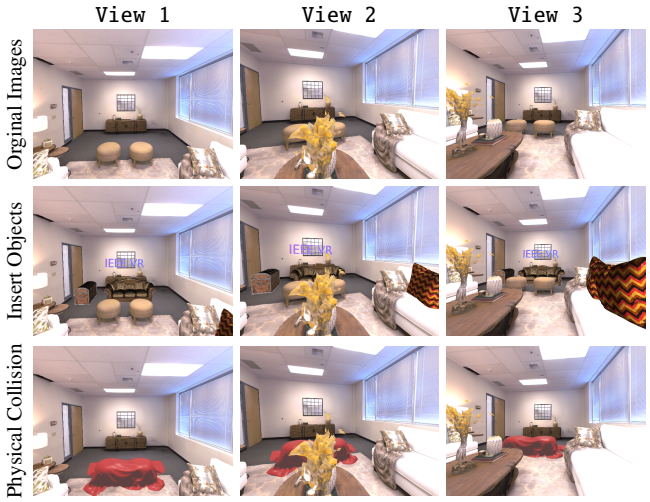


Figure 9: The AR demo on the scene `Room 0` of Replica [49]. We show two kinds of augmented reality applications. The first row shows the original RGB image. The second row shows the images with our inserted AR objects and `IEEE VR` text logo. The third row shows the physical collision between our reconstructed geometry and the virtual blanket.

In this part, we show the applications of our visual localization approach in the AR field. Our approach can not only estimate the 6-DoF pose of the camera in the reconstructed scene but also perform high-quality novel view rendering based on the Gaussian primitives. Therefore, our method can be very suitable for AR applications. We show two different AR applications in Fig. 9 on scene `Room 0` from the Replica dataset. **(1) Insert Objects**: we virtual AR objects and `IEEE VR` text logo into real scene. **(2) Physical collision**: we place a virtual blanket in the scene and let it fall naturally to simulate physical collision between our reconstructed scene geometry and the virtual blanket. As can be seen, our approach can perform high-quality rendering and handle the collision and occlusion between real and virtual content very well. For more AR demo videos, please refer to our supplementary video.

5 Conclusion and Limitation

In this paper, we propose SplatLoc, an efficient and novel visual localization approach based on the 3D Gaussian primitives, which is more suitable for AR/VR than traditional localization methods. Specifically, to compress the scene model for localization, we learn an unbiased 3D descriptor field for reconstructed Gaussian primitives, which is more accurate than previous alpha-blending approaches. Then, we propose a salient 3D landmark selection algorithm to select more informative primitives for visual localization based on the saliency score of Gaussian primitives, which can reduce the memory and runtime requirements for mobile devices. Besides, we propose an effective regularization term for key Gaussian primitives to avoid anisotropic shapes and reduce geometric errors, which can lead to stable localization performance improvement. Extensive experiments on two commonly used datasets have shown the effectiveness and application of our proposed system.

Currently, our proposed approach has two limitations. The first one is that we need depth information or sparse point clouds to reconstruct the scene. Our approach is based on 3DGS [18], which needs point cloud to initialize the position of each Gaussian primitive. The second one is that our method cannot be used for large outdoor scenes, which will increase the number of parameters. In the future, we will try to use the visual foundation model (*e.g.*, DepthAnything [58]) to estimate the depth of the RGB image, which can be viewed as prior to replacing the depth sensor and guide the scene reconstruction process. Besides, considering using a hierarchical representation approach [19] to extend our localization approach for large outdoor scenes.

REFERENCES

[1] R. Arandjelovic, P. Gronát, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5297–5307, 2016. 6

[2] A. Bergamo, S. N. Sinha, and L. Torresani. Leveraging structure from motion to learn discriminative codebooks for scalable landmark classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 763–770, 2013. 5

[3] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-aware learning of maps for camera localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2616–2625, 2018. 1, 2

[4] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021. 2

[5] D. Chen, H. Li, W. Ye, Y. Wang, W. Xie, S. Zhai, N. Wang, H. Liu, H. Bao, and G. Zhang. PGSR: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *arXiv preprint arXiv:2406.06521*, 2024. 2

[6] L. Chen, W. Chen, R. Wang, and M. Pollefeys. Leveraging neural radiance fields for uncertainty-aware visual localization. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024. 2, 6, 7, 8

[7] J. Chung, J. Oh, and K. M. Lee. Depth-regularized optimization for 3d gaussian splatting in few-shot images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 811–820, 2024. 2

[8] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt. BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration. *ACM Trans. Graph.*, 36(3):24:1–24:18, 2017. 4

[9] D. DeTone, T. Malisiewicz, and A. Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 224–236, 2018. 1, 2, 4, 5, 6

[10] T. Do, O. Miksik, J. DeGol, H. S. Park, and S. N. Sinha. Learning to detect scene landmarks for camera localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 5

[11] S. Dong, S. Wang, Y. Zhuang, J. Kannala, M. Pollefeys, and B. Chen. Visual localization via few-shot scene region classification. In *International Conference on 3D Vision*, pp. 393–402, 2022. 1, 2, 6, 7

[12] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 6

[13] X. Gao, X. Hou, J. Tang, and H. Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(8):930–943, 2003. 1

[14] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*, 2024. 2

[15] Y.-H. Huang, Y.-T. Sun, Z. Yang, X. Lyu, Y.-P. Cao, and X. Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4220–4230, 2024. 2

[16] J. T. Kajiya and B. V. Herzen. Ray tracing volume densities. In *SIGGRAPH*, pp. 165–174, 1984. 2

[17] A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A convolutional network for real-time 6-dof camera relocalization. In *IEEE International Conference on Computer Vision*, pp. 2938–2946, 2015. 1, 2

[18] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 1, 2, 3, 6, 10

[19] B. Kerbl, A. Meuleman, G. Kopanas, M. Wimmer, A. Lanvin, and G. Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics*, 43(4), 2024. 10

[20] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023. 2

[21] S. Kobayashi, E. Matsumoto, and V. Sitzmann. Decomposing nerf for editing via feature field distillation. *Advances in Neural Information Processing Systems*, 35:23311–23330, 2022. 2, 3

[22] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate $O(n)$ solution to the pnp problem. *Int. J. Comput. Vis.*, 81(2):155–166, 2009. 1, 2, 6

[23] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. 2

[24] H. Li, T. Fan, H. Zhai, Z. Cui, H. Bao, and G. Zhang. BDLoc: Global localization from 2.5D building map. In *IEEE International Symposium on Mixed and Augmented Reality*, pp. 80–89, 2021. 2

[25] H. Li, X. Yang, H. Zhai, Y. Liu, H. Bao, and G. Zhang. Vox-Surf: Voxel-based implicit surface representation. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–12, 2022. 2

[26] H. Li, H. Zhai, X. Yang, Z. Wu, Y. Zheng, H. Wang, J. Wu, H. Bao, and G. Zhang. Imtooth: Neural implicit tooth for dental augmented reality. *IEEE Trans. Vis. Comput. Graph.*, 29(5):2837–2846, 2023. 2

[27] X. Li, S. Wang, Y. Zhao, J. Verbeek, and J. Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11980–11989, 2020. 1, 2, 6, 7, 8

[28] J. Lin, Z. Li, X. Tang, J. Liu, S. Liu, J. Liu, Y. Lu, X. Wu, S. Xu, Y. Yan, et al. Vastgaussian: Vast 3d gaussians for large scene reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5166–5175, 2024. 2

[29] Y. Lin, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T. Lin. inerf: Inverting neural radiance fields for pose estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1323–1330, 2021. 2

[30] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353. 1998. 4

[31] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 1, 2

[32] Q. Ma, Y. Li, B. Ren, N. Sebe, E. Konukoglu, T. Gevers, L. V. Gool, and D. P. Paudel. ShapeSplat: A large-scale dataset of gaussian splats and their self-supervised pretraining. *arXiv preprint arXiv:2408.10906*, 2024. 2

[33] Q. Ma, D. P. Paudel, A. Chhatkuli, and L. Van Gool. Continuous pose for monocular cameras in neural implicit representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5291–5301, 2024. 2

[34] H. Matsuki, R. Murai, P. H. Kelly, and A. J. Davison. Gaussian splatting slam. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18039–18048, 2024. 2, 6

[35] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995. 2, 3

[36] M. Mera-Trujillo, B. Smith, and V. Fragoso. Efficient scene compression for visual-based localization. In *International Conference on 3D Vision (3DV)*, pp. 1–10, 2020. 5

[37] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoor-

thi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020. 1, 2

[38] Y. Ming, X. Yang, W. Wang, Z. Chen, J. Feng, Y. Xing, and G. Zhang. Benchmarking neural radiance fields for autonomous robots: An overview. *arXiv preprint arXiv:2405.05526*, 2024. 2

[39] A. Moreau, N. Piasco, D. Tsishkou, B. Stanciulescu, and A. de La Fortelle. LENS: localization enhanced by nerf synthesis. In *Conference on Robot Learning*, vol. 164, pp. 1347–1356, 2021. 2

[40] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 2, 4

[41] T. Ng, A. Lopez-Rodriguez, V. Balntas, and K. Mikolajczyk. Reassessing the limitations of cnn methods for camera pose regression. In *International Conference on 3D Vision*, 2021. 1, 6, 7

[42] S. Niedermayr, J. Stumpfegger, and R. Westermann. Compressed 3d gaussian splatting for accelerated novel view synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10349–10358, 2024. 2

[43] P. Papantonakis, G. Kopanas, B. Kerbl, A. Lanvin, and G. Drettakis. Reducing the memory footprint of 3d gaussian splatting. *ACM on Computer Graphics and Interactive Techniques*, 7(1):1–17, 2024. 2

[44] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister. Langsplat: 3d language gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20051–20060, 2024. 2, 3, 4, 8

[45] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021. 2

[46] J. Revaud, C. R. de Souza, M. Humenberger, and P. Weinzaepfel. R2D2: reliable and repeatable detector and descriptor. In *Advances in Neural Information Processing Systems*, pp. 12405–12415, 2019. 1, 2

[47] P. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From Coarse to Fine: Robust hierarchical localization at large scale. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12716–12725, 2019. 1, 2

[48] J.-C. Shi, M. Wang, H.-B. Duan, and S.-H. Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5333–5343, 2024. 2, 3, 4

[49] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, and R. Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 6, 7, 9

[50] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison. iMAP: Implicit mapping and positioning in real-time. In *IEEE/CVF International Conference on Computer Vision*, pp. 6209–6218, 2021. 1, 4

[51] V. Tschernezki, I. Laina, D. Larlus, and A. Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In *2022 International Conference on 3D Vision (3DV)*, pp. 443–453, 2022. 3

[52] M. Turkulainen, X. Ren, I. Melekhov, O. Seiskari, E. Rahtu, and J. Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. *arXiv preprint arXiv:2403.17822*, 2024. 2

[53] J. P. C. Valentin, A. Dai, M. Nießner, P. Kohli, P. H. S. Torr, S. Izadi, and C. Keskin. Learning to navigate the energy landscape. In *International Conference on 3D Vision*, pp. 323–332, 2016. 6, 7

[54] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20310–20320, 2024. 2

[55] H. Xiang, X. Li, X. Lai, W. Zhang, Z. Liao, K. Cheng, and X. Liu. Gaussianroom: Improving 3d gaussian splatting with sdf guidance and monocular cues for indoor scene reconstruction. *arXiv preprint arXiv:2405.19671*, 2024. 2

[56] Q. Xu, Z. Xu, J. Philip, S. Bi, Z. Shu, K. Sunkavalli, and U. Neumann. Point-nerf: Point-based neural radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5438–5448, 2022. 2

[57] C. Yan, D. Qu, D. Xu, B. Zhao, Z. Wang, D. Wang, and X. Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19595–19604, 2024. 2

[58] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024. 10

[59] L. Yang, R. Shrestha, W. Li, S. Liu, G. Zhang, Z. Cui, and P. Tan. Scenesqueezer: Learning to compress scene for camera relocalization. In *IEEE/CVF conference on computer vision and pattern recognition*, pp. 8259–8268, 2022. 5

[60] X. Yang, H. Li, H. Zhai, Y. Ming, Y. Liu, and G. Zhang. Vox-Fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *IEEE International Symposium on Mixed and Augmented Reality*, pp. 499–507, 2022. 2

[61] H. Yu, W. Ye, Y. Feng, H. Bao, and G. Zhang. Learning bipartite graph matching for robust visual localization. In *IEEE International Symposium on Mixed and Augmented Reality*, pp. 146–155, 2020. 1, 2

[62] H. Zhai, G. Huang, Q. Hu, G. Li, H. Bao, and G. Zhang. NIS-SLAM: Neural implicit semantic RGB-D SLAM for 3D consistent scene understanding. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–11, 2024. 2

[63] H. Zhai, H. Li, X. Yang, G. Huang, Y. Ming, H. Bao, and G. Zhang. Vox-fusion++: Voxel-based neural implicit dense tracking and mapping with multi-maps. *arXiv preprint arXiv:2403.12536*, 2024. 2

[64] B. Zhao, L. Yang, M. Mao, H. Bao, and Z. Cui. PNeRFLoc: Visual localization with point-based neural radiance fields. In *Conference on Artificial Intelligence*, pp. 7450–7459, 2024. 1, 2, 6, 7, 8

[65] S. Zhi, T. Laidlow, S. Leutenegger, and A. J. Davison. In-place scene labelling and understanding with implicit scene representation. In *IEEE/CVF International Conference on Computer Vision*, pp. 15838–15847, 2021. 6

[66] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross. Ewa volume splatting. In *Proceedings Visualization*, pp. 29–538, 2001. 3