BevSplat: Resolving Height Ambiguity via Feature-Based Gaussian Primitives for Weakly-Supervised Cross-View Localization

Qiwei Wang[1], Shaoxun Wu[1], Yujiao Shi[1]

[1]ShanghaiTech University, Shanghai, China. Correspondence to: Yujiao Shi <shiyj2@shanghaitech.edu.cn>

Abstract

This paper addresses the problem of weakly supervised cross-view localization, where the goal is to estimate the pose of a ground camera relative to a satellite image with noisy ground truth annotations. A common approach to bridge the cross-view domain gap for pose estimation is Bird's-Eye View (BEV) synthesis. However, existing methods struggle with height ambiguity due to the lack of depth information in ground images and satellite height maps. Because a single 2D pixel could represent points at various depths and heights, its true 3D position is ambiguous. Previous solutions either assume a flat ground plane or rely on complex models, such as cross-view transformers. We propose BevSplat, a novel method that resolves height ambiguity by using feature-based Gaussian primitives. Each pixel in the ground image is represented by a 3D Gaussian with semantic and spatial features, which are synthesized into a BEV feature map for relative pose estimation. We validate our method on the widely used KITTI and VIGOR datasets, which include both pinhole and panoramic query images. Experimental results show that BevSplat significantly improves localization accuracy over prior approaches. Our code is available at https://github.com/wangqww/BevSplat.

1 Inotroduction

Cross-view localization, the task of estimating the pose of a ground camera with respect to a satellite or aerial image, is a critical problem in computer vision and remote sensing. This task is especially important for applications such as autonomous driving, urban planning, and geospatial analysis, where accurately aligning ground-level and satellite views is crucial. However, it presents significant challenges due to the inherent differences in scale, perspective, and environmental context between ground-level images and satellite views.

To navigate these complexities, particularly the common difficulty of acquiring precise ground-truth (GT) camera locations at scale, weakly supervised learning [1, 2] has recently emerged as a promising paradigm. In this setting, models are trained using only noisy annotations, such as approximate camera locations with errors potentially reaching tens of meters, which adds another layer of complexity to the task. Nevertheless, the primary advantage of weak supervision lies in its ability to leverage less labor-intensive data collection, making it a more scalable and practical approach for many real-world applications where extensive precise annotations are infeasible.

A key strategy to address cross-view localization is Bird's-Eye View (BEV) synthesis [3–5, 1, 6], which generates a bird's-eye view representation from the ground-level image. The BEV image can then be compared directly to a satellite image, facilitating relative pose estimation. However, existing methods often rely on Inverse Perspective Mapping (IPM), which assumes a flat ground plane [1, 6],

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

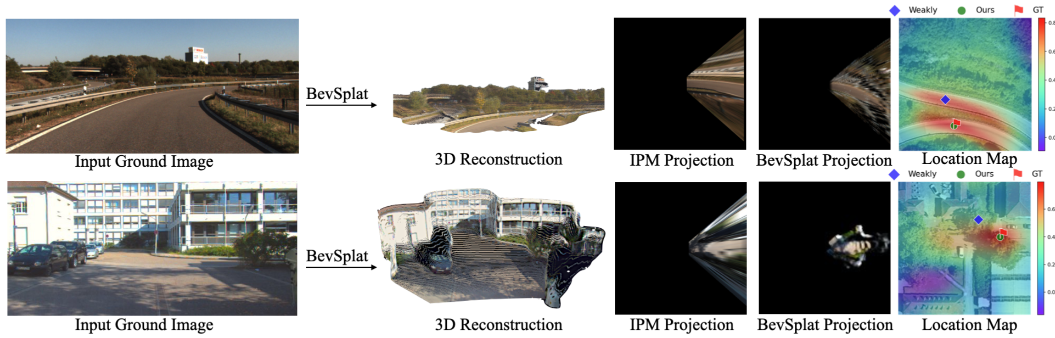

Figure 1: Our BevSplat cross-view localization process begins by using a depth prediction network on a single ground-level image to acquire its depth map. This depth map is then employed for 3D reconstruction into Gaussian splats, which are finally projected to a Bird's-Eye View (BEV). In comparison to the Inverse Perspective Mapping (IPM) approach, our method demonstrates improved recovery of BEV curves, more effective handling of building occlusions, and enhanced practical localization performance.

or on high-complexity models like cross-view transformers [3–5] to address height ambiguity, the challenge of resolving the elevation difference between the ground and satellite views.

The flat terrain assumption used in IPM leads to the loss of critical scene information above the ground plane and introduces distortions for objects farther from the camera, as shown in Fig. 1. On the other hand, while cross-view transformers are effective at handling distortions and objects above the ground plane, they are computationally expensive. Furthermore, in weakly supervised settings, noisy ground camera pose annotations provide weak supervision, making it difficult for high-complexity models like transformers to converge [1], ultimately leading to suboptimal localization performance.

In this paper, we propose BevSplat to address these challenges. BevSplat generates feature-based 3D Gaussian primitives for BEV synthesis. Unlike previous 3D Gaussian Splatting (3DGS) [7] methods that rely on color-based representations, we represent each pixel in the ground-level image as a 3D Gaussian with semantic and spatial features. These Gaussians are associated with attributes such as position in 3D space, scale, rotation, and density, which are synthesized into a BEV feature map using a visibility-aware rendering algorithm that supports anisotropic splatting. This approach enables us to handle height ambiguity and complex cross-view occlusions, improving the alignment between the ground-level image and the satellite view for more accurate pose estimation, without the need for expensive depth sensors or complex model architectures.

We validate our approach on the widely used KITTI and VIGOR datasets, where the former localizes images captured by pin-hole cameras, and the latter aims to localize panoramic images, demonstrating that the proposed BevSplat significantly outperforms existing techniques in terms of localization accuracy in various localization scenarios.

2 Related Work

2.1 Cross-view Localization

Cross-view localization, which aligns ground-level images with satellite imagery, has evolved from image retrieval to fine-grained pose estimation. Early approaches framed this as an image retrieval task, using metric learning to match ground queries to satellite image slices [8–12]. While modern transformers have improved retrieval performance, practical application remains challenging [13, 14]. Many recent methods adopt a coarse-to-fine pipeline. This typically involves a coarse retrieval step [15] followed by fine-grained (pixel-level) localization to identify the precise camera pose [16–18, 3, 19, 5, 20–22]. A key limitation of these methods is their reliance on precise, GPS-based training data, which is often prone to inaccuracies. To overcome this, weakly supervised settings have been proposed to learn from noisy pose annotations [1, 2]. These weakly supervised settings differ in their assumptions. [2] assumes the availability of GT labels in a source domain and access to cross-view pairs in the target domain. In contrast, [1] addresses a more challenging scenario where source domain GT labels are unavailable and no target domain pairs are accessible. In this work, we tackle the same task setting as [1].

2.2 Bird's-Eye View Synthesis

BEV synthesis, which generates bird's-eye view images from ground-level perspectives, has been widely applied to cross-view localization. While LiDAR and Radar sensors offer high accuracy for localization tasks [23–26], their high cost limits their use. For camera-only systems, multi-camera setups are commonly employed [27–30], primarily focusing on tasks like segmentation and recognition. In localization, methods like Inverse Perspective Mapping (IMP) assume a flat ground plane for BEV synthesis [1, 6, 20, 18], which can be overly simplistic for complex environments. Transformer-based models address these challenges but struggle with weak supervision and noisy pose annotations [3–5]. While methods such as [31, 32] also employ feature Gaussians for the image-to-BEV transformation, they typically benefit from rich depth and semantic information afforded by multi-sensor setups. While effective in some contexts, they face limitations in resource-constrained, real-world scenarios. In stark contrast, our approach is constrained to rely exclusively on weakly supervised signals derived purely from images, presenting a considerably more challenging task.

2.3 Sparse-View 3D Reconstruction

In our method, we adopt algorithms similar to 3D reconstruction to represent ground scenes. Sparse-view 3D reconstruction has been a major focus of the community. Nerf-based approaches [33] and their adaptations [34] have shown the potential for single-view 3D reconstruction, though their application is limited by small-scale scenes and high computational cost. Recent works using diffusion models [35–37] and 3D Gaussian representations [7, 38–40], as well as transformer- and Gaussian-based models[41, 42], have achieved sparse-view 3D reconstruction on a larger scale, but the complexity of these models still restricts their use due to computational demands. Approaches like [43–45] leverage pre-trained models to directly generate Gaussian primitives, avoiding the limitations of complex models while enabling scene reconstruction from sparse views. We apply such methods to single-view reconstruction, achieving high-accuracy cross-view localization.

3 Method

In this paper, we address cross-view localization by aligning ground-level and satellite images under weak supervision, where initial ground camera locations are only approximate. Our objective is to accurately estimate camera pose from these noisy priors by leveraging Gaussian primitives, which effectively manage height ambiguity and enable efficient generation of Bird's-Eye View (BEV) feature maps. First, we employ an orientation prediction network analogous to that in G2SWeakly to align the orientations of the ground and satellite images. Subsequently, our BevSplat method lifts the ground image to 3D (Section 3.1) and projects the corresponding Feature Gaussians into the BEV perspective to render the ground-view BEV features (Section 3.2.1). Finally, these features are compared against the satellite features by computing a similarity score (Section 3.2.2).

3.1 Geometric Gaussian Primitives Generation

Inspired by 3D Gaussian Splatting (3DGS) [7], we represent the 3D scene as a collection of Gaussian primitives. Our generation process first establishes their initial geometry and appearance. Given the inherent difficulty of directly learning accurate depth in our weakly supervised framework, we utilize a pre-trained depth estimation model to predict per-pixel depth D_i from the ground-level image. The initial 3D coordinate μ_i for primitives associated with each pixel (u_i, v_i) is then determined from D_i and the specific camera model.

For pinhole cameras, μ_i is computed by back-projecting 2D image coordinates (u_i, v_i) using depth D_i and camera intrinsics K as $\mu_i = K^{-1} D_i [u_i, v_i, 1]^T$.

For panoramic cameras, where pixel coordinates (u_i, v_i) represent viewing angles (e.g., azimuth u_i and polar angle v_i), the initial 3D coordinate μ_i is obtained by scaling the depth D_i along a unit direction vector $\hat{\mathbf{d}}_i = [x_i, y_i, z_i]^T$. The components are defined as:

$$x_i = -\sin(v_i)\cos(u_i), \quad y_i = -\cos(v_i), \quad z_i = -\sin(v_i)\sin(u_i). \tag{1}$$

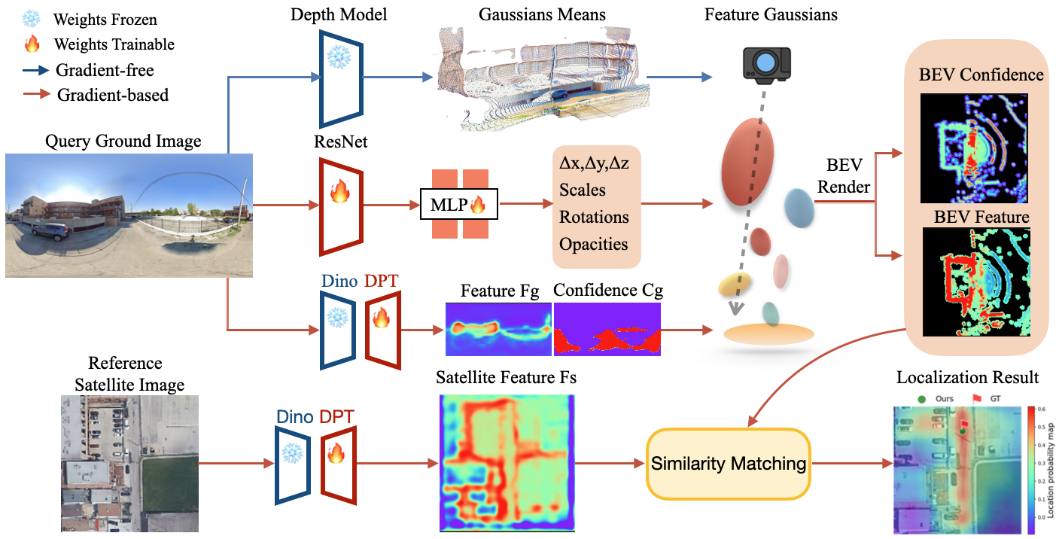

Figure 2: **BevSplat Framework Overview.** Query ground image Gaussian primitive initialization involves: (1) A pre-trained depth model for initial 3D positions (μ_i). (2) A ResNet and MLP to predict offsets ($\Delta \mathbf{p}_k$), scale ($\mathbf{S}_k$), rotation ($\mathbf{R}_k$), and opacity (O_k). (3) A DPT-fine-tuned DINOv2 for extracting semantic features ($\mathbf{f}_i$) and confidences (c_i), which are then bound to these Gaussians. These feature Gaussians are subsequently rendered into BEV feature and confidence maps. Satellite image features are extracted using an identical DINOv2-DPT backbone (note: weights are shared for KITTI but differ for VIGOR, similar to G2SWeakly [1]). Localization is achieved by matching satellite features with the rendered query BEV features via cosine similarity within a sliding window.

The initial 3D coordinate is thus $\mu_i = D_i \hat{\mathbf{d}}_i$.

Subsequently, a ResNet [46] extracts local features $\mathbf{f}^i_{\text{loc}}$ for each pixel i from the ground-level image. These features serve as input to a multi-layer perceptron (MLP [47]), denoted F_{gs}, which predicts attributes for $N_p = 3$ distinct Gaussian primitives originating from each pixel. The predicted attributes for each of these N_p Gaussian primitives include positional offsets $\Delta \mathbf{p}_k = (\Delta x_k, \Delta y_k, \Delta z_k)$ relative to μ_i, an anisotropic scale $\mathbf{S}_k$, a rotation quaternion $\mathbf{R}_k$, and an opacity value O_k. Predicting multiple primitives per pixel enhances single-image representation density. The collection of parameters G_i for these N_p primitives from pixel i, incorporating their final 3D positions, is:

$$G_i = \{(\mathbf{S}_k, \mathbf{R}_k, O_k, \mu_i + \Delta x_k, \mu_i + \Delta y_k, \mu_i + \Delta z_k)\}_{k=1}^{N_p}. \tag{2}$$

Here, k indexes the N_p primitives associated with that pixel, the set of parameters $\{(\mathbf{S}_k, \mathbf{R}_k, O_k, \Delta x_k, \Delta y_k, \Delta z_k)\}_{k=1}^{N_p}$ is the direct output of $F_{gs}(\mathbf{f}^i_{\text{loc}})$ and $\mathbf{f}^i_{\text{loc}}$ is the ResNet feature vector for pixel i. This process yields an initial set of geometric and appearance-based Gaussian primitives. These primitives are subsequently enriched with semantic features for localization, as detailed next.

3.2 Feature-based Gaussian Primitives for Relative Pose Estimation

For robust semantic feature extraction from ground and satellite images, inspired by [43, 48, 44], we fine-tune a pre-trained DINOv2 [49] model augmented with a DPT [50] module. From the ground image, this pipeline yields a feature map $\mathbf{F}_g \in \mathbb{R}^{H_g \times W_g \times C}$ and a confidence map $\mathbf{C}_g \in \mathbb{R}^{H_g \times W_g \times 1}$. The confidence map $\mathbf{C}_g$, derived from $\mathbf{F}_g$ via an additional convolutional layer and a sigmoid activation, assigns lower weights to dynamic objects (e.g., vehicles) and higher weights to static elements (e.g., road surfaces), indicating feature reliability for localization. For the predominantly static satellite image, we solely extract its feature map $\mathbf{F}_s \in \mathbb{R}^{H_s \times W_s \times C}$.

3.2.1 BEV Feature Rendering

The extracted ground features $\mathbf{F}_g$ and confidences $\mathbf{C}_g$ are then bound to the Gaussian primitives generated as described in Section 3.1. Specifically, each of the $N_p = 3$ Gaussian primitives originating from a ground image pixel i is augmented with the corresponding per-pixel feature vector $\mathbf{f}_i$ (sampled from $\mathbf{F}_g$) and confidence score c_i (from $\mathbf{C}_g$). All N_p primitives derived from the same pixel i thus share identical $\mathbf{f}_i$ and c_i values, effectively embedding semantic information and its reliability into the 3D representation.

Next, viewing the scene from a BEV perspective (camera directed downwards), the features $\mathbf{f}_b$ and confidences c_b bound to each Gaussian primitive b are rendered onto a 2D plane. This differentiable α-blending process, analogous to RGB rendering in 3DGS [7], yields the BEV feature map $\mathbf{F}_{BEV}$ and confidence map $\mathbf{C}_{BEV}$:

$$\mathbf{F}_{BEV} = \sum_{b=1}^{\mathcal{N}_\mathcal{G}} \mathbf{f}_b \alpha_b T_b, \quad \mathbf{C}_{BEV} = \sum_{b=1}^{\mathcal{N}_\mathcal{G}} c_b \alpha_b T_b, \tag{3}$$

where primitives $b \in \{1, \dots, \mathcal{N}_\mathcal{G}\}$ are sorted by depth from the BEV camera. Here, $T_b = \prod_{j=1}^{b-1}(1 - \alpha_j)$, $\mathcal{N}_\mathcal{G}$ is the total number of ground-image Gaussian primitives, α_b is the opacity of primitive b, and $\mathbf{f}_b, c_b$ are its bound feature vector and confidence score (inherited from its source pixel), respectively.

3.2.2 Pose Estimation via Confidence-Guided Similarity Learning

The location probability map $\mathbf{P}(u, v)$, representing the similarity between satellite image features $\mathbf{F}_s(u, v)$ and confidence-weighted ground BEV features $\mathbf{C}_{BEV}\mathbf{F}_{BEV}$, is computed as:

$$\mathbf{P}(u, v) = \frac{\langle \mathbf{F}_s(u, v), \mathbf{C}_{BEV}\mathbf{F}_{BEV} \rangle}{\|\mathbf{F}_s(u, v)\| \cdot \|\mathbf{C}_{BEV}\mathbf{F}_{BEV}\|}. \tag{4}$$

Here, $\mathbf{F}_s$ denotes the satellite image features, while $\mathbf{F}_{BEV}$ and $\mathbf{C}_{BEV}$ are the BEV features and confidence map derived from the ground image, respectively. $\| \cdot \|$ signifies the L_2 norm.

Supervision. Following [1], a deep metric learning objective supervises the network. For each query ground image, we compute location probability maps: $\mathbf{P}_{\text{pos}}$ against its positive satellite image and $\mathbf{P}_{\text{neg,idx}}$ against each of the M negative satellite images. The weakly supervised loss $\mathcal{L}_{\text{Weakly}}$ aims to maximize the peak of $\mathbf{P}_{\text{pos}}$ while minimizing the peak for each $\mathbf{P}_{\text{neg,idx}}$:

$$\mathcal{L}_{\text{Weakly}} = \frac{1}{M} \sum_{\text{idx}=1}^{M} \log\left(1 + e^{\alpha\left[\text{Peak}(\mathbf{P}_{\text{neg,idx}}) - \text{Peak}(\mathbf{P}_{\text{pos}})\right]}\right), \tag{5}$$

where the hyperparameter α (set to 10) controls convergence speed.

If more accurate (though potentially noisy) location labels (x^*, y^*) are available during training (e.g., GPS with error up to $d = 5$ meters, where β is the ground resolution in m/pixel of $\mathbf{P}_{\text{pos}}$), an auxiliary loss $\mathcal{L}_{\text{GPS}}$ is introduced:

$$\mathcal{L}_{\text{GPS}} = \left| \text{Peak}(\mathbf{P}_{\text{pos}}) - \text{Peak}(\mathbf{P}_{\text{pos}}[x^* \pm d/\beta, y^* \pm d/\beta]) \right|. \tag{6}$$

This objective encourages the global peak of $\mathbf{P}_{\text{pos}}$ to align with the local peak probability found within the d-meter radius neighborhood of the noisy label (x^*, y^*).

The total optimization objective is then:

$$\mathcal{L}_{all} = \mathcal{L}_{\text{Weakly}} + \lambda_1 \mathcal{L}_{\text{GPS}}, \tag{7}$$

where $\lambda_1 = 1$ if noisy location labels are utilized during training, and $\lambda_1 = 0$ otherwise.

4 Experiments

In this section, we first describe the benchmark datasets and evaluation metrics for evaluating the effectiveness of cross-view localization models, followed by implementation details of our method. Subsequently, we compare our method with state-of-the-art approaches and conduct experiments to demonstrate the necessity of each component of the proposed method.

KITTI dataset. The KITTI dataset [51] consists of ground-level images captured by a forward-facing pinhole camera with a restricted field of view, complemented by aerial images [52], where each

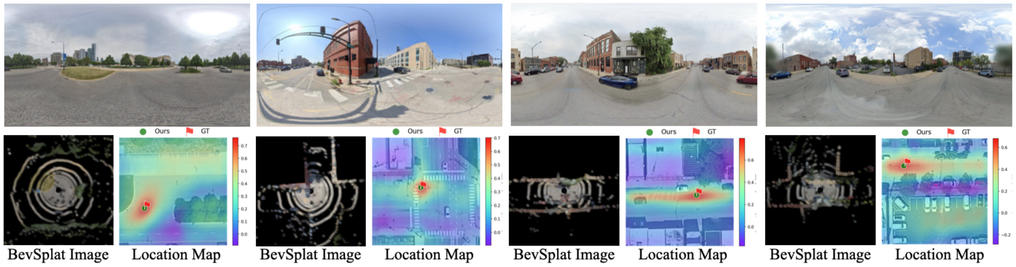

Figure 3: Visualization of the query ground image (up) and the estimated relative pose with respect to the satellite image (bottom right) on VIGOR dataset. The BEV image projected from the query ground image using the estimated Gaussian primitives is presented in the bottom left for each example.

aerial patch covers a ground area of approximately $100 \times 100\mathrm{m}^2$. The dataset includes a training set and two test sets(Same-Area and Cross-Area). For the Same-Area test set, the test query images are from the same geographical regions as the training set, but are not the same images. For the Cross-Area test set, the test images come from entirely new geographical regions that were not seen during training, testing the model's ability to generalize. The location search range of ground images is approximately $56 \times 56\mathrm{m}^2$, with an orientation noise of $\pm 10°$.

VIGOR dataset. The VIGOR dataset [15] includes geo-tagged ground panoramas and satellite images from four US cities: Chicago, New York, San Francisco, and Seattle. Each satellite patch spans $70 \times 70\mathrm{m}^2$ and is labeled positive if the ground camera is within its central $1/4$ region; otherwise, it is semi-positive. The dataset also has Same-Area and Cross-Area splits: Same-Area uses training and testing data from the same region, while Cross-Area splits training and testing between two separate city groups. We use only positive satellite images for all experiments, following [1].

Evaluation Metrics. For the KITTI dataset [51], we evaluated localization and orientation errors by calculating mean and median errors in meters and degrees, respectively. We also compute recall at thresholds of 1 m and 3 m for longitudinal (along the driving direction) and lateral (orthogonal to the driving direction) localization errors, as well as 1 ° and 3 ° for orientation errors. A localization is considered successful if the estimated position falls within the threshold of the ground truth, and an orientation is accurate if its error is within the angle threshold. For the VIGOR dataset [15], which does not provide driving direction information, we report mean and median errors as outlined in [1].

Visualization. We provide visualizations of the query images and localization results in Fig.3. For better clarity, we show the synthesized BEV image generated from our estimated Gaussian primitives at the bottom left of each example (though the model uses BEV feature maps for localization). Further qualitative results, including an analysis of failure cases, are provided in the supplementary material.

Implementation Details. For 3D point cloud generation from ground images, we employ specific depth estimation models: DepthAnythingV2 [53] for the pinhole camera images of the KITTI dataset [51], and UniK3D [54] for the panoramic images of the VIGOR dataset [15]. Our feature extractor for both ground and satellite images is a DINOv2 backbone [49], initialized with FiT weights [48], which is subsequently fine-tuned using an attached DPT module [50]. This extractor yields satellite feature maps with dimensions $(C, H, W) = (32, 128, 128)$. For ground images, initial feature maps of $(32, 64, 256)$ are extracted; these are then projected into BEV using our feature Gaussians projection, resulting in final ground BEV features also of dimensions $(32, 128, 128)$. Our model is trained using the AdamW optimizer [55] (weight decay 10^{-3}) and a OneCycleLR scheduler [56] with a cosine annealing strategy, where the learning rate peaks at 6.25×10^{-5}. We use a batch size of 8 on a single 4090 GPU, training for 8 epochs on KITTI and 14cf epochs on VIGOR.

4.1 Comparison with State-of-the-Art Methods

We compare our method with the latest state-of-the-art (SOTA) approaches, including supervised methods such as Boosting [4], VFA [20], CCVPE [57], HC-Net [6], and DenseFlow [18], all of which rely on ground-truth camera poses for supervision. We also compare with G2Sweakly [1], which uses only a satellite image and a corresponding ground image as input, similar to our setup.

Table 1: Comparison with the most recent state-of-the-art (SOTA) on KITTI (* denotes supervised learning algorithms). Our weakly supervised approach slightly outperforms SOTA supervised methods in Cross-Area evaluations.

Algorithms	λ_1	Test Area	localization		Lateral		Longitudinal		Azimuth			
			mean(m)↓	median(m)↓	d=1m↑	d=3m↑	d=1m↑	d=3m↑	$\theta = 1°$↑	$\theta = 3°$↑	mean(°)↓	median(°)↓
Boosting [4]*	-	Same-Area	12.08	11.42	76.44	96.34	23.54	50.57	99.10	**100.00**	-	-
VFA [20]*	-		10.74	10.51	51.17	-	5.19	-	49.85	96.98	1.40	1.00
CCVPE [57]*	-		1.22	0.62	97.35	**98.65**	77.13	**96.08**	77.39	99.47	0.67	0.54
HC-Net [6]*	-		**0.80**	0.50	**99.01**	-	**92.20**	-	91.35	99.84	0.45	0.33
DenseFlow [18]*	-		1.48	**0.47**	95.47	-	87.89	-	89.40	-	0.49	0.30
G2SWeakly [1]	0		12.03	8.10	59.58	85.74	11.37	31.94	**99.99**	**100.00**	**0.33**	**0.28**
Ours	0		5.82	2.85	60.04	91.54	24.06	56.82	**99.99**	**100.00**	**0.33**	**0.28**
G2SWeakly [1]	1		6.81	3.39	66.07	94.22	16.51	49.96	**99.99**	**100.00**	**0.33**	**0.28**
Ours	1		2.87	2.06	52.90	94.24	35.62	76.57	**99.99**	**100.00**	**0.33**	**0.28**
Boosting [4]*	-	Cross-Area	12.58	12.11	57.72	86.77	14.15	34.59	98.98	**100.00**	-	-
VFA [20]*	-		11.12	10.95	27.82	-	5.75	-	18.42	71.00	3.95	3.03
CCVPE [57]*	-		9.16	3.33	44.06	81.72	23.08	52.85	57.72	92.34	1.55	0.84
HC-Net [6]*	-		8.47	4.57	**75.00**	-	**58.93**	-	33.58	83.78	3.22	1.63
DenseFlow [18]*	-		7.97	3.52	54.19	-	23.10	-	43.44	-	2.17	1.21
G2SWeakly [1]	0		13.87	10.24	62.73	86.53	9.98	29.67	**99.99**	**100.00**	**0.33**	**0.28**
Ours	0		7.05	3.22	58.15	92.62	23.08	51.61	**99.99**	**100.00**	**0.33**	**0.28**
G2SWeakly [1]	1		12.15	7.16	64.74	86.18	11.81	34.77	**99.99**	**100.00**	**0.33**	**0.28**
Ours	1		**6.20**	**2.51**	51.45	**95.17**	27.41	**60.45**	**99.99**	**100.00**	**0.33**	**0.28**

KITTI. The comparison results on the KITTI dataset [51] are summarized in Table 1. Since our rotation estimator is inherited from G2Sweakly [1], the rotation estimation performance is identical between the two methods. However, our method significantly outperforms G2Sweakly [1] in terms of location estimation across almost all evaluation metrics, yielding substantial improvements in both longitudinal pose accuracy and the corresponding mean and median errors. This improvement can be attributed to the limitations of the IPM projection method used in G2Sweakly [1], which suffers from distortions in scenes that are far from the camera and fails to capture the details of objects above the ground plane.

Our feature-based Gaussian splatting for BEV synthesis effectively addresses these issues, leading to a notable enhancement in localization accuracy. Fig. 1 and Fig. 4 visualize the difference between the IPM projection and our proposed BEV synthesis method, clearly demonstrating that our projection technique resolves challenges such as occlusions caused by tall objects (e.g., buildings, trees, vehicles) and geometric distortions from curved roads. Furthermore, in cross-area evaluations, our method even surpasses supervised approaches (Boosting [4], VFA [20], CCVPE [57], HC-Net [6], and DenseFlow [18]) in terms of mean and median errors, showcasing the strong generalization ability of our approach and highlighting the potential of weakly supervised methods.

It is worth noting that the experimental results for VFA [20] are taken from the PIDLoc [22] paper. This is because the authors of VFA have adopted a setting in their prior works [58, 59] that aligns ground-truth poses to the satellite image center, which risks overfitting by biasing predictions towards the center. Since the official code for VFA [20] is not available, we attempted to reproduce their results and found that we could only achieve the reported performance by using this same setting. However, when we use the standard setting adopted by our method and other comparable works [4, 57, 6, 18, 1], our reproduced results are consistent with the VFA results reported in PIDLoc [22]. Therefore, for a fair comparison, we use the VFA [20] results from PIDLoc [22].

VIGOR. The comparison results on the VIGOR dataset [15] are presented in Table 2. Our method demonstrates a significant reduction in both mean and median errors compared to the baseline weakly supervised approach, G2SWeakly [1], across all evaluation scenarios. Furthermore, even when benchmarked against state-of-the-art fully supervised methods, our method maintains comparable performance in same-area evaluations while achieving notable improvements across most metrics in cross-area evaluations. This reduces the gap between weakly supervised and fully supervised methods, indicating that our approach generalizes effectively to diverse localization tasks, including both same-area and cross-area scenarios, as well as cases where the query images are either panoramic or captured using pinhole cameras.

Table 2: Comparison with the most recent state-of-the-art (SOTA) on VIGOR (* denotes supervised learning algorithms). Our weakly supervised approach achieves performance comparable to SOTA supervised methods in Same-Area evaluations and comprehensively surpasses them in Cross-Area evaluations.

Method	λ_1	Same-Area				Cross-Area			
		Aligned-orientation		Unknown-orientation		Aligned-orientation		Unknown-orientation	
		Mean(m)↓	Median(m)↓	Mean(m)↓	Median(m)↓	Mean(m)↓	Median(m)↓	Mean(m)↓	median(m)↓
Boosting [4]*	-	4.12	1.34	-	-	5.16	1.40	-	-
CCVPE [57]*	-	3.60	1.36	3.74	1.42	4.97	1.68	5.41	1.89
HC-Net [6]*	-	**2.65**	1.17	-	-	3.35	1.59	-	-
DenseFlow [18]*	-	3.03	**0.97**	4.97	1.90	5.01	2.42	7.67	3.67
G2SWeakly [1]	0	5.22	1.97	5.33	2.09	5.37	1.93	5.37	1.93
Ours	0	3.15	1.45	3.18	1.49	3.03	1.41	3.05	1.43
G2SWeakly [1]	1	4.19	1.68	4.18	1.66	4.70	1.68	4.52	1.65
Ours	1	2.87	1.58	**2.91**	**1.60**	**2.84**	**1.36**	**2.89**	**1.38**

Computation comparison. All evaluations of GPU memory usage were performed on an NVIDIA RTX 4090. Our model, which features a DINOv2 [49] backbone fine-tuned with our lightweight DPT network [50] (composed of a few CNN layers), requires considerably less memory during the training phase compared to G2SWeakly [1]. On the KITTI [51] dataset (batch size 8), our training phase uses only 9.2 GB of GPU memory, substantially less than the 22.7 GB required by G2SWeakly [1]. During inference, our model consumes 7.7 GB of GPU memory, while G2SWeakly [1] requires 7.2 GB as shown in Table 5. This increase 0.5 GB for our method is attributable to the larger DINOv2 backbone [49], representing a trade-off for its enhanced feature representation capabilities.

Table 3: BEV synthesis comparison on the KITTI dataset.

Rendering Method	λ_1	Same Area		Cross Area	
		Mean (m)↓	Median (m)↓	Mean (m)↓	Median (m)↓
IPM	0	9.02	5.54	9.97	6.35
Lift-Splat-Shoot [60]	1	16.14	13.94	17.74	14.51
OrienterNet [5]	0	15.59	13.68	16.15	13.80
Direct Projection	0	7.59	4.25	8.93	5.81
BevSplat (w/o OPT)	0	7.42	4.16	8.81	5.74
BevSplat (w/ OPT)	0	**5.82**	**2.85**	**7.05**	**3.22**
IPM	1	6.68	3.71	8.60	4.84
Lift-Splat-Shoot [60]	1	7.89	4.30	11.63	5.31
OrienterNet [5]	1	5.71	3.20	10.02	5.07
Direct Projection	1	7.59	4.25	8.93	5.81
BevSplat (w/o OPT)	1	4.37	3.21	7.86	4.57
BevSplat (w/ OPT)	1	**2.87**	**2.06**	**6.20**	**2.51**

Table 4: Ablation study on backbone module on the KITTI dataset.

Methods	λ_1	Same Area		Cross Area	
		Mean(m)↓	Median(m)↓	Mean(m)↓	Median(m)↓
Direct Train	0	17.74	15.61	17.59	15.71
LoRA [61]	0	16.29	14.48	17.05	14.7
DPT	0	5.82	2.85	7.05	3.22
Direct Train	1	14.32	12.43	17.28	15.14
LoRA [61]	1	13.58	11.79	16.81	14.63
DPT	1	2.87	2.06	6.20	2.51

Table 5: Comparison of resource consumption.

Method	Training Memory	Inference Memory	Inference Time
OrienterNet [5]	32.4	10.8	71
LSS [60]	26.1	8.3	85
G2SWeakly [1]	22.7	7.2	31
Ours	9.2	7.7	44

4.2 Ablation Study

Different BEV synthesis approaches. To validate the effectiveness of our BevSplat method, we compared it against two common BEV generation techniques: the Inverse Perspective Mapping (IPM) approach as utilized in [1], and direct projection of 3D point clouds. For a fair comparison, both baseline methods and our BevSplat employed the same DINOv2 backbone for feature extraction.

Table 6: Ablation study on backbone module on the KITTI dataset.

BackBone	λ_1	Same Area		Cross Area	
		Mean(m)↓	Median(m)↓	Mean(m)↓	Median(m)↓
VGG	0	8.77	4.53	9.91	5.80
DINOv1	0	7.68	4.01	9.16	4.67
DINOv2	0	7.04	3.45	8.37	4.21
DINOv2(FIT)	0	**5.82**	**2.85**	**7.05**	**3.22**
VGG	1	6.49	2.72	8.02	4.29
DINOv1	1	4.77	2.98	7.12	3.37
DINOv2	1	4.21	2.73	7.01	3.18
DINOv2(FIT)	1	**2.87**	**2.06**	**6.20**	**2.51**

The IPM projection method. This technique assumes all pixels in the ground-level image correspond to a flat plane at a real-world height of 0 meters. Consequently, while IPM can accurately represent flat road surfaces in BEV, it introduces significant distortions for any objects with non-zero elevations. These objects are typically stretched along the line of sight in the BEV. For instance, in the first row depicted in Fig. 4(b), vehicles' features appear elongated into regions not corresponding to their actual ground footprint. Similarly, in the second row Fig. 4(b), buildings' features are distorted and erroneously projected. A further limitation is that IPM typically projects only the lower portion of the ground image to BEV, discarding valuable information from the upper half. In contrast,

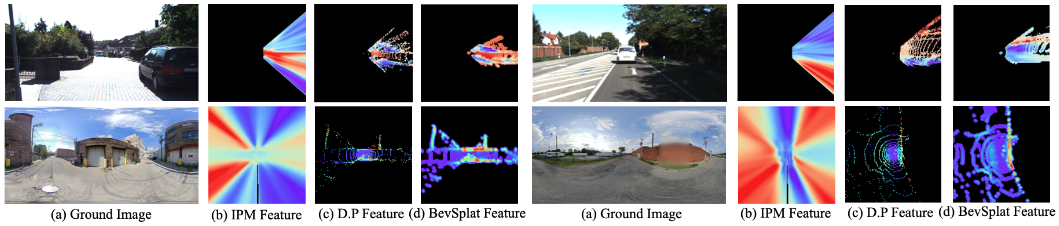

Figure 4: Visualization of query ground images (a), the corresponding BEV feature maps synthesized by IPM (b), by direct projection (c), and by the proposed BevSplat (d). The top two examples are from the KITTI dataset, while the bottom two are from the VIGOR dataset.

our BevSplat method is designed to leverage geometric information from the entire image more effectively.

Direct point cloud projection to BEV. While this approach can address some of the aforementioned IPM limitations, it introduces new challenges stemming from point cloud sparsity (visualized in Fig. 4(c)). This sparsity leads to BEV voids and discontinuous features, exacerbated by the lack of control over individual point attributes such as opacity, scale, and shape. Furthermore, unlike 3DGS [7] which utilizes α-blending, the simple top-down projection inherent in this method causes severe occlusion, leading to the loss of underlying feature information—an issue also evident in Fig. 4(c). Our ablation study (Table 3) confirms this inherent characteristic: BevSplat(w/o OPT) configured with non-optimizable Gaussian parameters (e.g., fixed opacity=1.0, scale=0.1, offsets=0, and no learned adjustments) performs comparably to direct point cloud projection, underscoring that point clouds can be seen as a degenerate form of 3DGS [7]. However, our full BevSplat(w/ OPT) formulation significantly improves upon this baseline by optimizing Gaussian opacity, scale, shape, and position. Guided by satellite imagery, this optimization process effectively mitigates the issues of point cloud projection, yielding coherent, feature-rich BEV representations and thereby enabling superior localization accuracy. As detailed in Table 3, BevSplat subsequently outperforms both IPM and Direct Projection methods.

Other depth-based re-sampling methods. We further compare our method with other BEV projection techniques that are also based on depth prediction, such as those used in Lift-Splat-Shoot [60] and OrienterNet [5] as detailed in Table 3, by adapting their projection modules into our framework. Although these approaches can generate a denser BEV and, like our method, leverage the full vertical information from the ground-view image—allowing them to perform well in same-area settings when guided by GPS labels—they fail to generalize to unseen environments. They tend to make erroneous guesses to fill in occluded regions, which explains their performance degradation in cross-area evaluations. Furthermore, they employ a simple weighted averaging for BEV projection, which is less accurate for handling vertical occlusions compared to BevSplat's principled alpha blending. Finally, both methods utilize a complex $h \times w \times d$ depth representation to perform an attention-based sum over ground features. This implicit, high-dimensional process incurs substantial computational and memory overhead to produce a denser BEV as shown in Table 5.

Foundation model backbone. To validate the effectiveness of fine-tuning a foundation model for extracting ground and satellite image features, we conducted ablation studies on the impact of different foundation models with their pre-trained weights, as well as the influence of various fine-tuning methods on the experimental results.

Impact of Different Foundation Models and Weights. Prioritizing robust outdoor generalization and effective 3D-relevant feature extraction for our foundation model, we selected DINOv2 [49] fine-tuned with the FiT method [48]. This model, which we term DINOv2(FiT), utilizes its *dinov2_base_fine* pre-trained weights renowned for these capabilities. To validate this choice and compare its efficacy against alternatives, our ablation study also evaluated VGG [62], DINOv1 [63], and the original DINOv2. All DINO-based backbones in this study (DINOv1, DINOv2, and DINOv2(FiT)) were subsequently further fine-tuned by us using a DPT-like module [50]. The ablation results (Table 6) validated our selection, as DINOv2(FiT), after our DPT-like fine-tuning, demonstrated superior performance among the evaluated backbones.

Impact of Different Fine-tuning Methods. Using DPT-like models is a common practice for 3D vision tasks; for example, VGGT [64] utilizes DPT [50] for point cloud reconstruction, depth estimation, and feature matching. Although we are the first to apply DPT-DINO for feature extraction in the specific sub-field of cross-view localization, our motivation is to similarly obtain features that are rich in 3D information. This is analogous to human navigation, where in addition to semantic information, an understanding of the real 3D scene is also crucial for localization. However, obtaining such 3D-aware features to bridge the significant ground-satellite domain gap is non-trivial. Simpler fine-tuning methods like direct end-to-end training or LoRA [61] fail, as they either lose crucial texture details or are not powerful enough to adapt the foundation model. We use a DPT-like module because its multi-scale feature fusion architecture is uniquely suited for this challenge. It successfully adapts the backbone by preserving both the low-level texture and high-level semantic information required for matching across these different domains as shown in Table 4.

Number of Gaussian primitives per pixel (N_p). The number of Gaussian primitives per pixel, N_p, also affects the resulting BEV feature quality. While an excessive N_p can complicate training and cause inter-primitive occlusions, an insufficient count leads to sparse and inadequate BEV representations. Our ablation study (Fig. 5) determined $N_p = 3$ to be optimal, offering the best balance between feature richness and model tractability.

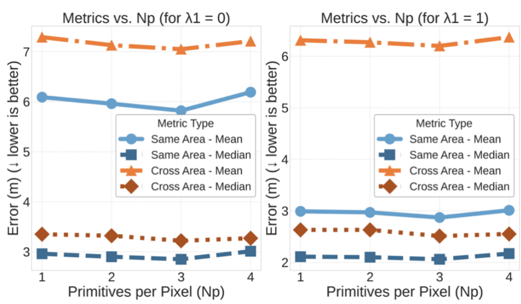

Figure 5: Ablation on primitives per pixel (N_p) The error is minimized when $N_p = 3$ on KITTI dataset.

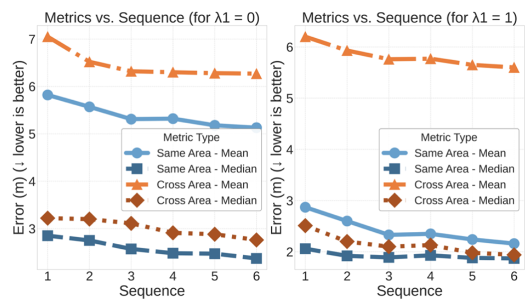

Figure 6: Location error with increasing sequence length on KITTI dataset.

4.3 Multi-Frame Localization

Beyond processing single ground-level images, our method extends to leveraging multiple frames from video sequences to enhance localization robustness, particularly in dynamic environments. Similar to CVLNet [65], given known inter-frame relative poses, our BevSplat technique projects features from these frames into a unified BEV. These multi-frame BEV features are then fused by a Transformer employing self-attention. We investigated this multi-frame capability using query video sequences comprising 1 to 6 frames. The results, presented in Fig. 6, demonstrate that localization performance consistently improves with an increasing number of frames in the sequence. This underscores the efficacy of our approach in leveraging temporal information from video data for enhanced localization robustness.

5 Conclusion

This paper has introduced a new approach for weakly supervised cross-view localization by leveraging feature-based 3D Gaussian primitives to address the challenge of height ambiguity. Unlike traditional methods that assume a flat ground plane or rely on computationally expensive models such as cross-view transformers, our method synthesizes a Bird's-Eye View (BEV) feature map using feature-based Gaussian splatting, enabling more accurate alignment between ground-level and satellite images. We have validated our approach on the KITTI and VIGOR datasets, demonstrating that our model achieves superior localization accuracy.

However, the inference speed of our method is currently constrained by the reconstruction and rendering overhead inherent to existing 3D Gaussian Splatting (3DGS) techniques. Future work will focus on developing faster reconstruction algorithms and more compact 3D Gaussian representations to enhance computational efficiency. Despite this current limitation, we believe that our approach provides a promising direction for scalable and accurate cross-view localization, paving the way for real-world applications in autonomous navigation, geospatial analysis, and beyond.

Acknowledge

The authors are grateful for the valuable comments and suggestions by the reviewers and ACs. This work was supported by NSFC (62406194), Shanghai Frontiers Science Center of Human-centered Artificial Intelligence (ShangHAI), MoE Key Laboratory of Intelligent Perception, HPC Platform of ShanghaiTech University and Human-Machine Collaboration (KLIP-HuMaCo). A part of the experiments of this work were supported by the core facility Platform of Computer Science and Communication, SIST, ShanghaiTech University.

References

[1] Y. Shi, H. Li, A. Perincherry, and A. Vora, "Weakly-supervised camera localization by ground-to-satellite image registration," in *European Conference on Computer Vision*. Springer, 2024, pp. 39–57.

[2] Z. Xia, Y. Shi, H. Li, and J. F. Kooij, "Adapting fine-grained cross-view localization to areas without fine ground truth," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

[3] F. Fervers, S. Bullinger, C. Bodensteiner, M. Arens, and R. Stiefelhagen, "Uncertainty-aware vision-based metric cross-view geolocalization," *arXiv preprint arXiv:2211.12145*, 2022.

[4] Y. Shi, F. Wu, A. Perincherry, A. Vora, and H. Li, "Boosting 3-dof ground-to-satellite camera localization accuracy via geometry-guided cross-view transformer," *arXiv preprint arXiv:2307.08015*, 2023.

[5] P.-E. Sarlin, D. DeTone, T.-Y. Yang, A. Avetisyan, J. Straub, T. Malisiewicz, S. R. Bulò, R. Newcombe, P. Kontschieder, and V. Balntas, "Orienternet: Visual localization in 2d public maps with neural matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 632–21 642.

[6] X. Wang, R. Xu, Z. Cui, Z. Wan, and Y. Zhang, "Fine-grained cross-view geo-localization using a correlation-aware homography estimator," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[7] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[8] T.-Y. Lin, S. Belongie, and J. Hays, "Cross-view image geolocalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 891–898.

[9] K. Regmi and A. Borji, "Cross-view image synthesis using conditional gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3501–3510.

[10] Y. Shi, L. Liu, X. Yu, and H. Li, "Spatial-aware feature aggregation for image based cross-view geo-localization," in *Advances in Neural Information Processing Systems*, 2019, pp. 10 090–10 100.

[11] L. Liu and H. Li, "Lending orientation to neural networks for cross-view geo-localization," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[12] Y. Shi, X. Yu, L. Liu, T. Zhang, and H. Li, "Optimal feature transport for cross-view image geo-localization." in *AAAI*, 2020, pp. 11 990–11 997.

[13] H. Yang, X. Lu, and Y. Zhu, "Cross-view geo-localization with layer-to-layer transformer," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 009–29 020, 2021.

[14] S. Zhu, M. Shah, and C. Chen, "Transgeo: Transformer is all you need for cross-view image geo-localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 1162–1171.

[15] S. Zhu, T. Yang, and C. Chen, "Vigor: Cross-view image geo-localization beyond one-to-one retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3640–3649.

[16] Y. Shi and H. Li, "Beyond cross-view image retrieval: Highly accurate vehicle localization using satellite image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 010–17 020.

[17] Z. Xia, O. Booij, M. Manfredi, and J. F. Kooij, "Visual cross-view metric localization with dense uncertainty estimates," in *European Conference on Computer Vision*. Springer, 2022, pp. 90–106.

[18] Z. Song, J. Lu, Y. Shi *et al.*, “Learning dense flow field for highly-accurate cross-view camera localization,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[19] T. Lentsch, Z. Xia, H. Caesar, and J. F. Kooij, “Slicematch: Geometry-guided aggregation for cross-view pose estimation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 225–17 234.

[20] S. Wang, C. Nguyen, J. Liu, Y. Zhang, S. Muthu, F. A. Maken, K. Zhang, and H. Li, “View from above: Orthogonal-view aware cross-view localization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 843–14 852.

[21] Z. Xia and A. Alahi, “Fg ^{2}: Fine-grained cross-view localization by fine-grained feature matching,” *arXiv preprint arXiv:2503.18725*, 2025.

[22] W. Lee, J. Park, D. Hong, C. Sung, Y. Seo, D. Kang, and H. Myung, “Pidloc: Cross-view pose optimization network inspired by pid controllers,” *arXiv preprint arXiv:2503.02388*, 2025.

[23] Y. Qin, C. Wang, Z. Kang, N. Ma, Z. Li, and R. Zhang, “Supfusion: Supervised lidar-camera fusion for 3d object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22 014–22 024.

[24] A. W. Harley, Z. Fang, J. Li, R. Ambrus, and K. Fragkiadaki, “Simple-bev: What really matters for multi-sensor bev perception?” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2759–2765.

[25] Z. Lin, Z. Liu, Z. Xia, X. Wang, Y. Wang, S. Qi, Y. Dong, N. Dong, L. Zhang, and C. Zhu, “Rcbevdet: Radar-camera fusion in bird’s eye view for 3d object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 928–14 937.

[26] Z. Liu, J. Hou, X. Ye, T. Wang, J. Wang, and X. Bai, “Seed: A simple and effective 3d detr in point clouds,” in *European Conference on Computer Vision*. Springer, 2025, pp. 110–126.

[27] L. Reiher, B. Lampe, and L. Eckstein, “A sim2real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird’s eye view,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–7.

[28] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai, “Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers,” *arXiv preprint arXiv:2203.17270*, 2022.

[29] J. Yang, E. Xie, M. Liu, and J. M. Alvarez, “Parametric depth based feature representation learning for object detection and segmentation in bird’s-eye view,” *ICCV*, 2023.

[30] P.-E. Sarlin, E. Trulls, M. Pollefeys, J. Hosang, and S. Lynen, “Snap: Self-supervised neural maps for visual positioning and semantic understanding,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 7697–7729, 2023.

[31] F. Chabot, N. Granger, and G. Lapouge, “Gaussianbev: 3d gaussian representation meets perception models for bev segmentation,” in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2025, pp. 2250–2259.

[32] S.-W. Lu, Y.-H. Tsai, and Y.-T. Chen, “Gaussianlss–toward real-world bev perception: Depth uncertainty estimation via gaussian splatting,” *arXiv preprint arXiv:2504.01957*, 2025.

[33] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[34] Y. Hong, K. Zhang, J. Gu, S. Bi, Y. Zhou, D. Liu, F. Liu, K. Sunkavalli, T. Bui, and H. Tan, “Lrm: Large reconstruction model for single image to 3d,” *arXiv preprint arXiv:2311.04400*, 2023.

[35] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” 2021.

[36] X. Ze, B. Zhu, Z. Song, J. Lu, and Y. Shi, “Satdreamer360: Multiview-consistent generation of ground-level scenes from satellite imagery,” 2025. [Online]. Available: https://arxiv.org/abs/2506.00600

[37] X. Ze, Z. Song, Q. Wang, J. Lu, and Y. Shi, “Controllable satellite-to-street-view synthesis with precise pose alignment and zero-shot environmental control,” 2025. [Online]. Available: https://arxiv.org/abs/2502.03498

[38] Y. Cai, H. Zhang, K. Zhang, Y. Liang, M. Ren, F. Luan, Q. Liu, S. Y. Kim, J. Zhang, Z. Zhang *et al.*, "Baking gaussian splatting into diffusion denoiser for fast and scalable single-stage image-to-3d generation," *arXiv preprint arXiv:2411.14384*, 2024.

[39] J. Zhou, W. Zhang, and Y.-S. Liu, "Diffgs: Functional gaussian splatting diffusion," *arXiv preprint arXiv:2410.19657*, 2024.

[40] Y. Mu, X. Zuo, C. Guo, Y. Wang, J. Lu, X. Wu, S. Xu, P. Dai, Y. Yan, and L. Cheng, "Gsd: View-guided gaussian splatting diffusion for 3d reconstruction," in *European Conference on Computer Vision*. Springer, 2025, pp. 55–72.

[41] Y. Chen, M. Mihajlovic, X. Chen, Y. Wang, S. Prokudin, and S. Tang, "Splatformer: Point transformer for robust 3d gaussian splatting," 2024. [Online]. Available: https://arxiv.org/abs/2411.06390

[42] H. Jiang, L. Liu, T. Cheng, X. Wang, T. Lin, Z. Su, W. Liu, and X. Wang, "Gausstr: Foundation model-aligned gaussian transformer for self-supervised 3d spatial understanding," *arXiv preprint arXiv:2412.13193*, 2024.

[43] S. Zhou, H. Chang, S. Jiang, Z. Fan, Z. Zhu, D. Xu, P. Chari, S. You, Z. Wang, and A. 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*, 2024, pp. 21 676–21 685.

[44] C. Wewer, K. Raj, E. Ilg, B. Schiele, and J. E. Lenssen, "latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction," in *European Conference on Computer Vision*. Springer, 2025, pp. 456–473.

[45] S. Szymanowicz, E. Insafutdinov, C. Zheng, D. Campbell, J. F. Henriques, C. Rupprecht, and A. Vedaldi, "Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image," *arXiv preprint arXiv:2406.04343*, 2024.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[47] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[48] Y. Yue, A. Das, F. Engelmann, S. Tang, and J. E. Lenssen, "Improving 2d feature representations by 3d-aware fine-tuning," in *European Conference on Computer Vision*. Springer, 2025, pp. 57–74.

[49] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[50] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 179–12 188.

[51] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[52] Y. Shi, X. Yu, L. Liu, D. Campbell, P. Koniusz, and H. Li, "Accurate 3-dof camera geo-localization via ground-to-satellite image matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[53] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," *arXiv preprint arXiv:2406.09414*, 2024.

[54] L. Piccinelli, C. Sakaridis, M. Segu, Y.-H. Yang, S. Li, W. Abbeloos, and L. Van Gool, "Unik3d: Universal camera monocular 3d estimation," *arXiv preprint arXiv:2503.16591*, 2025.

[55] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[56] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Artificial intelligence and machine learning for multi-domain operations applications*, vol. 11006. SPIE, 2019, pp. 369–386.

[57] Z. Xia, O. Booij, and J. F. Kooij, "Convolutional cross-view pose estimation," *arXiv preprint arXiv:2303.05915*, 2023.

[58] S. Wang, Y. Zhang, A. Perincherry, A. Vora, and H. Li, "View consistent purification for accurate cross-view localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8197–8206.

[59] S. Wang, Y. Zhang, A. Vora, A. Perincherry, and H. Li, "Satellite image based cross-view localization for autonomous vehicle," *arXiv preprint arXiv:2207.13506*, 2022.

[60] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," 2020. [Online]. Available: https://arxiv.org/abs/2008.05711

[61] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

[62] K. Simonyan, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

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

[64] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, "Vggt: Visual geometry grounded transformer," 2025. [Online]. Available: https://arxiv.org/abs/2503.11651

[65] Y. Shi, X. Yu, S. Wang, and H. Li, "Cvlnet: Cross-view semantic correspondence learning for video-based camera localization," in *Asian Conference on Computer Vision*. Springer, 2022, pp. 123–141.

[66] D. Charatan, S. L. Li, A. Tagliasacchi, and V. Sitzmann, "pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 457–19 467.

[67] 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.

[68] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, "Zoedepth: Zero-shot transfer by combining relative and metric depth," 2023. [Online]. Available: https://arxiv.org/abs/2302.12288

[69] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," 2024. [Online]. Available: https://arxiv.org/abs/2406.09414

[70] S. Gasperini, N. Morbitzer, H. Jung, N. Navab, and F. Tombari, "Robust monocular depth estimation under challenging conditions," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2023. [Online]. Available: http://dx.doi.org/10.1109/ICCV51070.2023.00751

[71] G. Bradski, "Learning opencv: Computer vision with the opencv library," *O'REILLY google schola*, vol. 2, pp. 334–352, 2008.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: We emphasize our main contributions and the scope of this work in the abstract and further detail them in the Introduction (Section 1).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

Justification: A detailed discussion of the limitations of our current approach is provided in the Conclusion(Section 5). Furthermore, additional failure cases are presented in the supplementary material for a comprehensive understanding.

Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

   Justification: Our paper does not include mathematical derivations related to formal proofs.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Details about the reproducibility of our experimental results are provided in the Implementation Details of Experiments section (Section 4).

Guidelines:

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

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

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

   Answer: [No]

   Justification: We will release all the code and data later.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We specify all the training and test details in the Experiments section (Section 4)

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: While quantitative results are provided, a formal analysis of their statistical significance (e.g., through error bars or hypothesis testing) is not included in this submission but is identified as an important area for future refinement.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Details of the computational resources employed are provided in the 'Computation Comparison' subsection of the Experiments section (see Section 4.1).

Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have adhered to the Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss both the positive and negative societal impacts of this work in our Conclusion section (see Section 5) and in the supplementary material.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All relevant prior work has been appropriately cited, with references primarily located in our Related Work section (see Section 2) and Experiments section (see Section 4). Furthermore, the use of any open-source code in this study adheres to the terms and policies of their respective licenses, including all requirements for attribution and acknowledgment.

Guidelines:

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

13. **New assets**

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

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

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

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

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [NA]

    Justification: We exclusively used a Large Language Model (LLM) for writing this paper.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Carefule Explanation of the Weakly Supervised Setup

We focus on a weakly-supervised setting because precise GPS data is often unavailable in the real world. To do this, we adopt the weakly-supervised setup from G2SWeakly [1], which defines two scenarios:

λ=0: the error of the location labels for ground images in the training dataset is the same as the error that the model aims to refine during deployment. For example, the error of location labels for ground images in the training data set is +/- 20m. During testing, the model is also given a location of query images with error up to 20m and aim to reduce this error.

λ=1: relatively more accurate location labels for ground images in the training data are available than the poses we aim to refine during employment. For example, the model was trained with images whose location labels have an error of +/- 5m. During testing, the query images have an initial location estimate with errors up to 20m, and the model aim to reduce this error.

B Concepts of "Height Ambiguity".

We use the term "height ambiguity" to describe the challenge of projecting a 2D ground-level image to a Bird's-Eye View (BEV). Because a single 2D pixel could represent points at various depths and heights, its true 3D position is ambiguous.

Methods like IPM resolve this ambiguity by assuming a flat ground plane. This introduces significant errors for any object with height, causing the characteristic distortions and smearing we aim to solve. We will clarify this definition in our revision

C Clarification on Occlusion Handling in the Differentiable Blending Process

Forward Pass: Following Equation 3, we sort all contributing Gaussians by depth and render them from front to back.

Backward Pass: The process is fully differentiable, allowing the loss to guide how the scene should be structured. For the feature vector (f_b): The gradient for the b-th Gaussian is computed as:

$$\frac{\partial L_{\text{all}}}{\partial f_b} = \frac{\partial L_{\text{all}}}{\partial F_{\text{BEV}}} \cdot T_b \cdot \alpha_b \tag{8}$$

The learning signal is scaled by transmittance (T_b) and opacity (α_b). This means the features of the most visible (least occluded) and most solid Gaussians are prioritized for updates.

For the opacity (α_b): The gradient of the b-th Gaussian is computed as:

$$\frac{\partial L_{\text{all}}}{\partial \alpha_b} = \frac{\partial L_{\text{all}}}{\partial F_{\text{BEV}}} \cdot T_b \cdot (f_b - f_b^{\text{accum}}) \tag{9}$$

The gradient depends on the difference between the current Gaussian's feature (f_b) and the accumulated features behind it (f_b^{accum}). This trains the model to make a Gaussian opaque if it is needed to hide a conflicting background, effectively learning to form solid, occluding surfaces.

In short, this mechanism is directly analogous to how the original 3DGS handles RGB colors, and we have repurposed it to optimize feature representations for localization.

D Robustness to Localization Errors

We evaluate the robustness of our method to varying levels of initial localization error. As shown in Table 7, localization performance improves significantly as the initialization error decreases.

E Ablation Study on Gaussian Primitive Offset and Scale

This ablation study investigates our method's sensitivity to the maximum offset and maximum scale of Gaussian Primitives. For each parameter, we evaluate values from the set {0.3, 0.5, 1.0}. The results,

Table 7: Performance comparison under different location error settings on KITTI dataset.

Location Error (m^2)	$\boldsymbol{\lambda_1}$	Same Area		Cross Area	
		Mean(m) ↓	Median(m) ↓	Mean(m) ↓	Median(m) ↓
56 × 56	0	5.82	2.85	7.05	3.22
	1	2.87	2.06	6.20	2.51
28 × 28	0	3.27	2.28	3.60	2.47
	1	2.43	1.94	3.31	2.21

presented in Table 8, demonstrate relatively stable performance across these configurations. Optimal performance is observed when both the maximum offset and scale are set to 0.5; consequently, these are adopted as their default values.

Table 8: Ablation study on max_offset and max_scaleon KITTI dataset.

Max_Offset(m)	Max_Scale(m)	$\boldsymbol{\lambda_1}$	Same Area		Cross Area	
			Mean(m) ↓	Median(m) ↓	Mean(m) ↓	Median(m) ↓
0.3	0.3	0	6.16	2.89	7.36	3.20
0.5	0.5	0	**5.82**	**2.85**	**7.05**	3.22
1.0	1	0	6.00	2.95	7.06	**3.16**
0.3	0.3	1	3.42	2.28	6.83	2.53
0.5	0.5	1	**2.87**	**2.06**	**6.20**	**2.51**
1.0	1	1	3.28	2.30	6.52	2.57

F Ablation Study on the Number of Gaussian Primitives Per Pixel(N_p)

We conducted an ablation study on the number of Gaussian primitives per pixel (N_p) across both the KITTI and VIGOR datasets to validate our design choice. The results for KITTI are presented in Figure 5 of our main paper, and the new results for the VIGOR dataset are provided below Table 9.

Table 9: Ablation study on the number of sampled points, N_p.

N_p	λ_1	Same Area		Cross Area	
		Mean(m) ↓	Median(m) ↓	Mean(m) ↓	Median(m) ↓
1	0	3.05	1.71	2.97	1.71
2	0	2.98	1.67	2.94	1.65
3	0	**2.96**	**1.62**	**2.90**	**1.65**
4	0	3.03	1.67	2.91	1.68
1	1	2.62	1.47	2.71	1.42
2	1	2.59	1.41	2.67	1.40
3	1	**2.57**	**1.40**	**2.63**	**2.54**
4	1	2.59	1.42	2.65	1.38

The results on VIGOR are consistent with our findings on KITTI: performance is optimal when =3. As we discuss in our paper:

- Using too few primitives can limit the model's ability to fill gaps in sparse regions.
- Using too many primitives can make training more difficult.

This finding is also consistent with prior work. Our design was inspired by PixelSplat [66], which similarly found Np=3 to be a robust and effective setting across multiple datasets. Therefore, we conclude that Np=3 is a well-justified hyperparameter that should be generally applicable.

G Applicability to Multi-Frame Localization Tasks

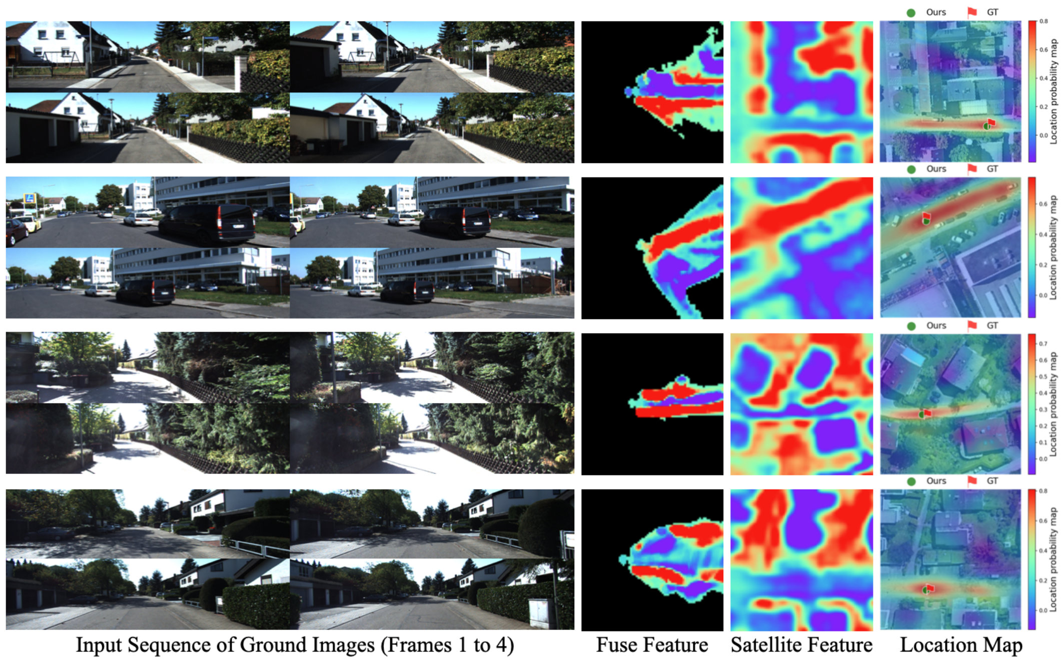

Figure 7: Visualization of multi-frame localization results on the KITTI dataset, achieved using our BevSplat-based approach. This demonstrates the aggregation of information over time and the filtering of dynamic elements.

As detailed in the main paper, our quantitative analysis of a CVLNet-based multi-frame fusion method [65] demonstrated progressively improved performance with an increasing number of frames, confirming its efficacy for temporal sequence tasks. To complement these findings, Figure 7 provides a qualitative illustration.

This visualization highlights how our fusion strategy effectively aggregates richer contextual information from multiple video frames. Notably, the approach adeptly filters dynamic objects while prioritizing the preservation of static scene elements, which are crucial for robust cross-view localization. These qualitative insights further substantiate the effectiveness and generalizability of our proposed method in handling dynamic environments and leveraging temporal information for improved localization.

H Multi-frame Fusion Comparison: BevSplat vs. IPM and Direct Point Cloud Projection

To demonstrate our method's consistency and fusion capabilities, we have conducted a new multi-frame comparison against both IPM and direct point cloud projection baselines. The Table 10 below present the performance on the KITTI dataset using a sequence of six frames. The values in parentheses show the percentage improvement from fusing six frames over the single-frame results:

Our BevSplat framework demonstrates superior multi-frame fusion capabilities for the following reasons:

- Inverse Perspective Mapping (IPM): BEV representations generated via IPM are prone to significant artifacts and distortions, which fundamentally hinder effective temporal fusion.
- Direct Point Cloud: Projections of raw point clouds result in sparse representations that handle occlusions poorly, often causing background features to bleed through foreground objects. This issue is not resolved well by aggregating multiple frames.

Table 10: Mutil-frame fusion comparison.

Methods	Seq	λ_1	Same Area		Cross Area	
			Mean(m)	Median(m)	Mean(m)	Median(m)
G2SWeakly	6	0	8.65(↓4.1%)	5.22(↓5.7%)	9.41(↓4.6%)	6.01(↓5.3%)
Direct Projection	6	0	7.05(↓7.1%)	3.86(↓9.2%)	8.15(↓8.7%)	5.31(↓8.6%)
Ours	6	0	5.01(↓13.9%)	2.27(↓20.4%)	6.09(↓13.6%)	2.71(↓15.8%)
G2SWeakly	6	1	6.25(↓6.4%)	3.38(↓8.9%)	8.18(↓4.9%)	4.32(↓10.7%)
Direct Projection	6	1	3.85(↓13.1%)	2.91(↓11.6%)	7.18(↓9.6%)	3.96(↓14.7%)
Ours	6	1	2.01(↓30.0%)	1.77(↓14.1%)	5.23(↓15.6%)	1.94(↓22.7%)

- Our Method: In contrast, BevSplat utilizes adaptive Gaussian primitives to create a dense, coherent BEV that faithfully represents the road topology. This provides a robust foundation for multi-frame fusion, as shown in Fig. 9 of our supplement.

I Sensitivity to Depth Prediction Quality and Failure Cases

We evaluate our framework's performance with three different depth foundation models: DepthAnythingv1 [67], ZoeDepth [68], and DepthAnythingv2 [69] in Table 11:

Table 11: Ablation study on different depth estimation methods.

Method	λ_1	Same Area		Cross Area	
		Mean(m)↓	Median(m)↓	Mean(m)↓	Median(m)↓
DepthAnythingV1 [67]	0	5.91	2.84	7.21	3.25
ZoeDepth [68]	0	5.84	2.86	7.14	3.22
DepthAnythingV2 [69]	0	5.82	2.85	7.05	3.22
DepthAnythingV1 [67]	1	2.97	2.11	6.28	2.52
ZoeDepth [68]	1	2.91	2.03	6.21	2.54
DepthAnythingV2 [69]	1	2.87	2.06	6.20	2.51

The results demonstrate that while a more accurate depth model improves performance, the overall system is not highly sensitive to the choice of different depth estimators. This robustness stems from our end-to-end differentiable design, which optimizes the initial 3D Gaussian positions during training, compensating for minor discrepancies between different depth priors. Our framework can seamlessly leverage future advancements in monocular depth estimation. We will include this analysis in our paper.

J Robustness in Adverse Environmental Conditions

To test our method's robustness, we generated synthetic Rain, Fog, and Night data for KITTI (following the methodology of Robust-Depth [70]). This allows for a controlled comparison against the G2SWeakly baseline under challenging conditions.

The results in Table 12 show that while both methods are affected by adverse conditions, our approach demonstrates greater relative robustness. The reason lies in how each method handles corrupted input:

- IPM-based methods like G2SWeakly project visual artifacts (e.g., rains, fog) directly onto the BEV, creating severe geometric distortions that corrupt the final representation.
- In contrast, our method, while starting with a less accurate depth map in these conditions, still preserves a stable underlying 3D structure. It avoids the stretching errors of IPM and is better able to ignore atmospheric noise, leading to a more graceful degradation in performance.

Table 12: Performance comparison under different weather conditions.

Method	Weather	λ_1	Same Area		Cross Area	
			Mean(m) ↓	Median(m) ↓	Mean(m) ↓	Median(m) ↓
G2SWeakly	Origin	0	9.02	5.54	13.97	10.24
	Rain	0	16.45	13.29	18.44	16.52
	Fog	0	12.82	9.8	15.47	12.03
	Night	0	14.42	11.31	17.25	14.66
Ours	Origin	0	5.82	2.85	7.05	3.22
	Rain	0	8.61	4.78	10.03	5.69
	Fog	0	6.60	3.23	8.27	4.29
	Night	0	7.59	4.11	10.77	6.16
G2SWeakly	Origin	1	6.68	3.71	12.15	7.16
	Rain	1	16.82	13.48	19.45	17.42
	Fog	1	10.19	5.48	15.67	12.53
	Night	1	11.56	7.43	17.72	16.53
Ours	Origin	1	2.87	2.06	6.20	2.51
	Rain	1	8.64	3.94	11.12	6.34
	Fog	1	4.21	2.50	8.21	3.43
	Night	1	7.95	3.32	10.39	7.09

K Coordinate System

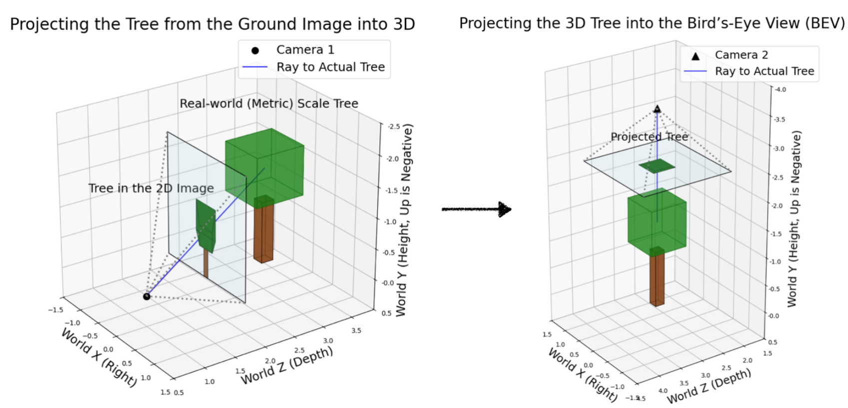

Figure 8: BevSplat Geometry Projection Overview. Our method is a two-stage geometry projection. *Left panel (Stage 1):* We reconstruct the 3D scene from ground-level images using their associated depth information, illustrated by converting a tree from a ground-level image to its 3D representation. *Right panel (Stage 2):* The reconstructed 3D scene is then projected into the Bird's-Eye View (BEV).

Our methodology employs a world coordinate system consistent with the OpenCV convention [71], as depicted in Figure 8. This is a right-handed system where, from the camera's viewpoint, the +X axis extends to its right, the +Y axis points downwards, and the +Z axis aligns with its forward viewing direction. Consequently, the upward direction corresponds to the -Y axis.

In the 3D reconstruction stage, a point cloud is generated from the input images. This is achieved by back-projecting pixels, using their depth information, along the initial camera's viewing direction (defined as the +Z axis of this coordinate system).

Subsequently, for Bird's-Eye View (BEV) projection, an aerial perspective is simulated. A virtual camera is conceptually positioned at a nadir viewpoint—looking directly downwards—above the reconstructed 3D scene. Given our coordinate system where the +Y axis points downwards, this BEV camera is located at a Y-coordinate that is numerically smaller than those of the scene's primary content (thus representing a higher altitude). It views along the +Y direction (downwards). The BEV is then formed by orthographically projecting the 3D point cloud onto the world's XZ-plane (which effectively serves as the ground plane) along this +Y viewing axis.

L Qualitative Results

Robust Performance in Complex Scenarios: As illustrated in the first two rows of images in Figure 9 and Figure 10, our method demonstrates robust localization performance across a variety of challenging scenarios, such as road intersections, curved road sections, and areas with significant occlusions from roadside trees, as validated on the `KITTI` and `VIGOR` datasets. This proficiency is primarily attributed to our approach's enhanced capabilities in: (1) effectively handling visual occlusions caused by buildings; (2) establishing and leveraging more accurate geometric relationships within the scene; and (3) optimally fusing features pertinent to the vertical spatial arrangement of elements, such as trees and road surfaces, between ground-level and aerial (*e.g.*, satellite) views. Consequently, our method achieves promising localization results in these complex environments, underscoring its effectiveness in tackling real-world complexities.

Limitations in Feature-Scarce Environments: Conversely, as illustrated in the last two rows of images in Figure 9 and Figure 10, in specific scenarios such as long, straight road segments that lack distinctive visual features, our method exhibits a comparative reduction in localization accuracy. The primary reason for this limitation is that in the absence of salient visual landmarks, the deep learning network, when attempting to match the ground-level view to the satellite imagery, may assign similar matching probabilities or confidence scores to multiple plausible locations within the satellite map. This multi-modal matching outcome leads to localization ambiguity, making it difficult for the network to make a unique, high-precision positioning decision.

M On the Benefits in Supervised vs. Weakly-Supervised Settings

Although our paper focused on the weakly-supervised setting, our framework also demonstrates strong performance with full supervision. To illustrate this, we trained our model and the G2SWeakly baseline in a fully supervised setting on the KITTI dataset. The results are in Table 13:

Table 13: Supervised vs. weakly-supervised settings.

Method	$\lambda 1$	Same Area		Cross Area	
		Mean(m) ↓	Median(m) ↓	Mean(m) ↓	Median(m) ↓
G2SWeakly(Supervised)	-	6.32	3.15	12.2	8.33
Ours(Supervised)	-	2.07	1.12	6.75	3.03
Ours(Weakly Supervised)	0	5.82	2.85	7.05	3.22
Ours(Weakly Supervised)	1	2.61	2.06	6.20	2.51

As the results show, while switching to a fully supervised setting, our model is highly competitive, in the challenging cross-area task. However, we found that our weakly-supervised model (λ_1=1) achieves even better cross-area performance than our own fully-supervised version. This suggests that precise supervision may cause overfitting to the training domain's biases, which is contrary to our goal of building a more generalizable system.

Therefore, our paper's focus on the weakly-supervised setting is twofold. First, it addresses the practical challenge that high-quality GPS data is often unavailable in the real world. Second, it is the setting where our method paradoxically achieves its best and most robust generalization performance.

N Limitations and Future Works

As discussed in the main paper's conclusion, a current limitation of our BevSplat method, which renders Bird's-Eye View (BEV) perspectives based on 3D Gaussian Splatting [7], is its computational speed compared to Inverse Perspective Mapping (IPM). For instance, on an NVIDIA RTX 4090 GPU, BevSplat requires 14 ms to generate a single BEV image. In contrast, IPM, which utilizes

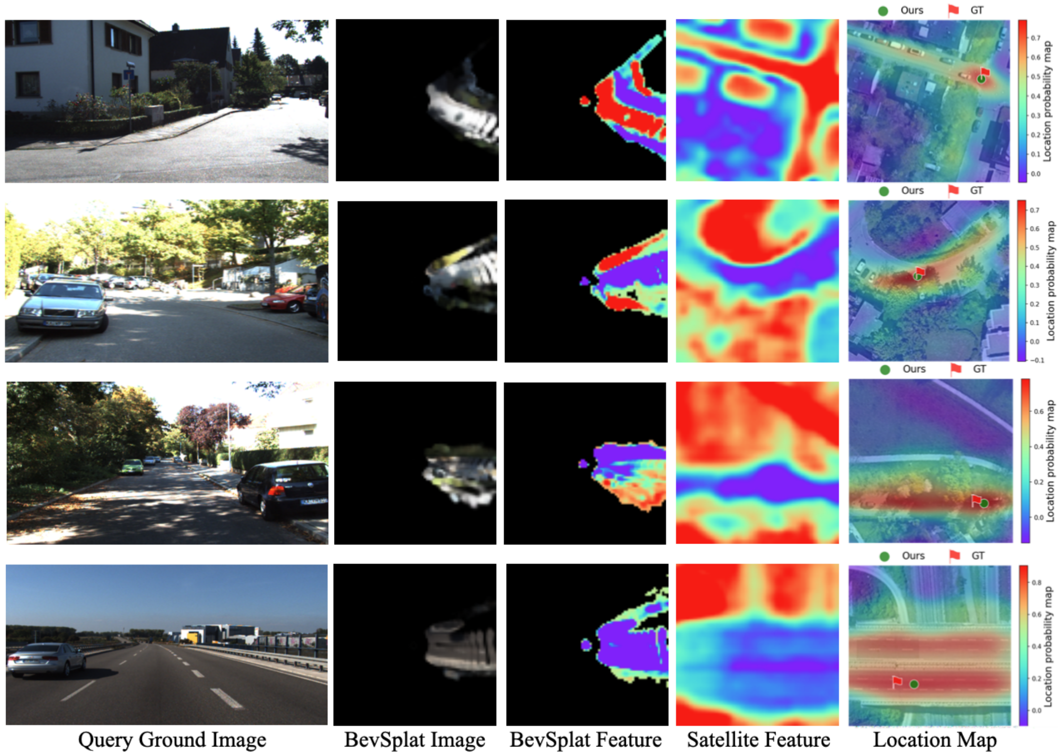

Figure 9: Qualitative results for BevSplat-based single-image localization on KITTI. Top two rows: successful examples; bottom two rows: failure examples.

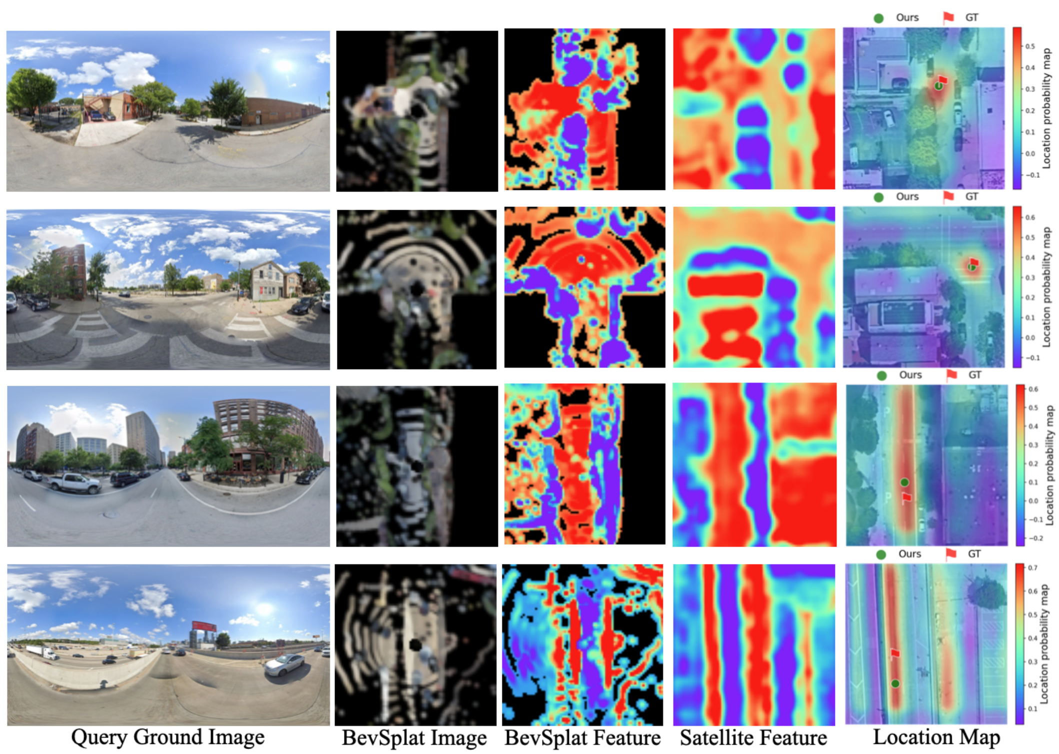

Figure 10: Qualitative results for BevSplat-based single-image localization on VIGOR. Top two rows: successful examples; bottom two rows: failure examples.

direct linear interpolation, can achieve this in 4ms. This performance disparity affects the overall inference speed of our model. Therefore, a significant direction for our future work is the exploration of faster and more compact Gaussian representations to address this bottleneck and enhance real-time applicability.

O Broader Impacts

Our work, BevSplat, addresses the critical demand for robust and accessible localization systems for mobile robots, such as drones and autonomous vehicles, particularly in scenarios where high-precision GPS is either unavailable or impractical due to cost or signal dependency. By leveraging computer vision, BevSplat delivers real-time, high-precision localization using only a monocular camera, or a camera augmented with low-cost, low-precision GPS. This significantly extends localization capabilities to GPS-denied or unreliable environments, a crucial step for the widespread adoption of autonomous systems.

To foster further research and collaboration within the community, we are committed to open-sourcing our complete codebase, training datasets, and pre-trained model weights on GitHub. This efficient implementation, which operates on a single NVIDIA RTX 4090 GPU, is provided as a resource for the research community. We encourage researchers to explore, build upon, and collaborate with us to advance this promising research direction, ultimately contributing to safer and more versatile autonomous navigation.