SPFSplatV2: Efficient Self-Supervised Pose-Free 3D Gaussian Splatting from Sparse Views

Ranran Huang, Krystian Mikolajczyk

Abstract—We introduce SPFSplatV2, an efficient feed-forward framework for 3D Gaussian splatting from sparse multi-view images, requiring no ground-truth poses during training and inference. It employs a shared feature extraction backbone, enabling simultaneous prediction of 3D Gaussian primitives and camera poses in a canonical space from unposed inputs. A masked attention mechanism is introduced to efficiently estimate target poses during training, while a reprojection loss enforces pixel-aligned Gaussian primitives, providing stronger geometric constraints. We further demonstrate the compatibility of our training framework with different reconstruction architectures, resulting in two model variants. Remarkably, despite the absence of pose supervision, our method achieves stateof-the-art performance in both in-domain and out-of-domain novel view synthesis, even under extreme viewpoint changes and limited image overlap, and surpasses recent methods that rely on geometric supervision for relative pose estimation. By eliminating dependence on ground-truth poses, our method offers the scalability to leverage larger and more diverse datasets. Code and pretrained models will be available on our project page: https://ranrhuang.github.io/spfsplatv2/.

Index Terms—Gaussian Splatting, novel view synthesis, self-supervised, pose-free, efficiency.

I. INTRODUCTION

Recent advancements in 3D reconstruction and novel view synthesis (NVS) have been driven by Neural Radiance Fields (NeRFs) [1] and 3D Gaussian splatting (3DGS) [2]. A standard training pipeline for novel view synthesis reconstructs a 3D scene from input views and optimizes it by aligning rendered novel views with ground-truth images [3]–[8].

State-of-the-art methods typically employ geometry-aware architectures by constructing cost volumes [3], [4], [6], leveraging epipolar transformers [5], or encoding camera poses using Plücker ray embeddings [9]–[11]. These approaches rely on camera poses estimated with Structure-from-Motion (SfM) [12] to reconstruct 3D scenes, as illustrated in Fig. 1 (a). However, acquiring camera poses from SfM is computationally expensive and often unreliable in sparse-view scenarios due to insufficient correspondences, limiting the applicability of these *pose-required* methods. To address this, recent research has focused on novel view synthesis under pose-free settings.

Existing pose-free methods reconstruct 3D scenes from unposed images by learning in a canonical space [7], [8], [13], [14], leveraging latent scene representations [15], [16], or jointly optimizing both context-view camera poses and

Ranran Huang, and Krystian Mikolajczyk are with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, United Kingdom. E-mail: {r.huang24; k.mikolajczyk}@imperial.ac.uk.

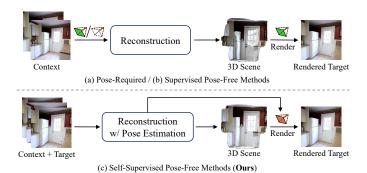


Fig. 1. Comparison of three typical **training** pipelines for sparse-view 3D reconstruction in novel view synthesis. For simplicity, the image rendering loss on the rendered target view is omitted. (a) Pose-required methods rely on ground-truth poses for both 3D scene reconstruction and target-view rendering. (b) Supervised pose-free methods requires no ground-truth poses for reconstruction but still rely on ground-truth poses for rendering loss. (c) Our self-supervised pose-free pipeline instead leverages estimated target poses to optimize 3D scene reconstruction from unposed images, thereby removing dependence on ground-truth poses during both training and inference.

3D scene representations [17]–[19]. Although these methods do not require accurate poses at inference, their training is still supervised by rendering losses given ground-truth poses at novel viewpoints, as shown in Fig. 1 (b). We therefore categorize these approaches as *supervised pose-free* methods. As a result, their reliance on training datasets with known camera poses restricts the scalability to large-scale real-world data without pose annotations.

This raises a critical question: Are ground-truth poses truly indispensable for optimizing 3D scenes during training? One solution is to use estimated poses at novel viewpoints, referred to as the self-supervised pose-free paradigm in Fig. 1 (c). However, this presents an inherent challenge: since the rendering loss intrinsically couples the learning of 3D scene geometry and camera poses, pose errors can degrade reconstruction quality, which further hampers pose estimation. Such mutual dependency creates a feedback loop that can potentially lead to unstable training or even divergence. Recent self-supervised pose-free approaches [20], [21] struggle to mitigate this issue primarily due to their use of separate and cascading modules for scene reconstruction and pose estimation, discouraging the learning of consistent feature representations across the two tasks and impairing geometric consistency. Consequently, these methods exhibit poor training stability, particularly under large viewpoint changes, and still lag far behind state-of-theart pose-required and supervised pose-free methods [5]–[7].

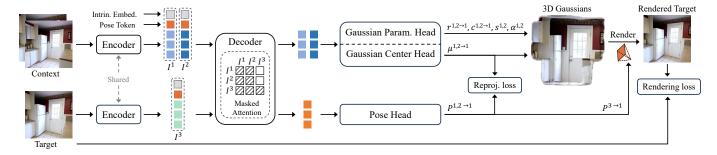


Fig. 2. Training pipeline of SPFSplatV2. A shared backbone with three specialized heads simultaneously predicts Gaussian centers, additional Gaussian parameters, and camera poses from unposed images in a canonical space, with the first input view as the reference. Encoder tokens, concatenated with a learnable pose token and an optional embedding of ground-truth intrinsics, are fed into the decoder, which employs masked attention to prevent context tokens from attending to target tokens, ensuring Gaussian reconstruction remains independent of target-view information. The 3D Gaussians are optimized via a rendering loss using the predicted target poses, while a reprojection loss enforces alignment between Gaussian centers and their corresponding pixels using the predicted context poses. By jointly optimizing Gaussians and camera poses, the pipeline enhances geometric consistency and improves reconstruction quality.

To address the challenge, we introduce SPFSplatV2, a self-supervised pose-free approach for 3D Gaussian splatting from unposed sparse views. As shown in Fig. 2, SPFSplatV2 employs a shared backbone for feature extraction with dedicated heads for predicting 3D Gaussian primitives and camera poses relative to a reference view. The unified backbone improves computational efficiency and facilitates joint feature learning for scene reconstruction and pose estimation, thereby enhancing geometric consistency and mitigating feedback instability. This is achieved by enabling 3D geometry to benefit from context-aware camera alignment and allowing pose predictions to leverage global scene context.

During training, in addition to context images, the target images are also incorporated as input for target pose estimation, enabling rendering losses at target views. To prevent information leakage from target images into the Gaussian reconstruction of context views, we introduce a masked attention mechanism, as shown in Fig. 2. In this design, context tokens attend only to context tokens, ensuring that 3D Gaussian reconstruction remains independent of target-view information. Conversely, target tokens attend to both context and target tokens, allowing the model to exploit global scene context for accurate target pose estimation. Finally, we complement the rendering loss with a reprojection loss that explicitly enforces alignment between the predicted Gaussians and their corresponding image pixels, imposing stronger geometric constraints and further enhancing training stability. In conclusion, we make the following key contributions:

- We propose SPFSplatV2, a feed-forward framework with masked attention that enables efficient and stable joint optimization of scene reconstruction and pose estimation from sparse unposed views, requiring no ground-truth poses during training and inference.
- SPFSplatV2 outperforms state-of-the-art pose-required, supervised pose-free, and self-supervised pose-free methods on both in-domain and out-of-domain novel view synthesis, demonstrating robustness under limited view overlap and extreme viewpoint changes. Despite relying solely on image supervision, its efficient feed-forward relative pose estimation surpasses most approaches that depend on geometric supervision.

 By eliminating the reliance on ground-truth poses during training, our method offers the scalability needed to leverage larger and more diverse datasets. Its effectiveness across different architectures further demonstrates the paradigm's broad compatibility.

This work substantially extends our previous method, SPF-Splat [22], with the key novelties summarized as follows:

- Methodological Improvements: Different from SPFSplat, which employs separate context-only and context-withtarget input branches to avoid target information leakage, we introduce a unified architecture with masked attention mechanism that reduces computational overhead and pose misalignment. Pose estimation is further improved with learnable pose tokens, which selectively attend to relative multi-view cues for more accurate camera inference. In addition, a multi-view dropout strategy enhances generalization across varying numbers and spatial distributions of context views.
- Architectural Compatibility: We demonstrate that our training paradigm is compatible with state-of-the-art reconstruction models. To this end, we develop two variants: SPFSplatV2, which follows a MASt3R-style [23] architecture (consistent with SPFSplat), and SPFSplatV2-L, which adopts the VGGT [24] architecture.
- Superior Performance: Extensive experiments show that SPFSplatV2 and SPFSplatV2-L achieve significant improvements over SPFSplat [22] and other state-of-the-art methods in novel view synthesis, cross-domain generalization and relative pose estimation.

II. RELATED WORK

A. Novel View Synthesis

NeRF [1] and 3DGS [2] have demonstrated strong performance in 3D reconstruction and novel view synthesis. Early methods rely on dense input views for per-scene optimization [25]–[28], whereas recent approaches focus on generalizable reconstruction from sparse-view images [3]–[11]. Typical NVS pipelines reconstruct 3D scenes from input views and optimize them by aligning synthesized images to ground-truth targets. Based on their dependence on ground-truth camera

poses during training and inference, existing methods can be grouped into pose-required, supervised pose-free, and self-supervised pose-free approaches, as illustrated in Fig. 1.

Pose-Required Methods reconstruct 3D scenes from images given accurate poses using geometry-aware architectures [3]–[6], [9]–[11]. For example, MVSNeRF [3] and MuRF [4] construct cost volumes for multi-view aggregation to reconstruct radiance fields, while MVSplat [6] uses cost volumes for depth estimation to reconstruct Gaussian primitives. Other strategies include epipolar transformers in pixelSplat [5] or encoding camera poses with Plücker ray embeddings [9]–[11]. Despite their effectiveness, these methods depend on SfM for precise camera poses, which is computationally expensive and often unreliable in sparse-view scenarios. Recent pose estimation techniques [23], [29]–[32] mitigate some issues but still struggle in low-overlap or texture-less settings. Consequently, pose-required methods remain impractical for unposed reconstruction during both training and inference.

Supervised Pose-Free Methods enable 3D reconstruction from unposed images, relaxing the need for camera poses at inference. Methods such as UpSRT [16] and UpFusion [15] encode unposed images into latent scene representations, while BARF [19], SPARF [33], DBARF [18], and CoPoNeRF [17] jointly optimize poses and NeRF representations. LEAP [13] and PF-LRM [14] leverage ViT architectures to define neural volumes in canonical camera coordinates. More recently, Splatt3R [8] predicts 3D Gaussians in a canonical space by regressing offsets to pointmaps from a frozen MASt3R [23], but requires depth supervision. NoPoSplat [7] removes this depth reliance and refines this pipeline by fine-tuning MASt3R and incorporating intrinsics to mitigate scale ambiguity. However, despite removing pose requirements at inference, these methods still depend on ground-truth poses during training via rendering losses [7], [8], [13]-[16], explicit pose supervision [17], or coarse pose initialization [19], [33], therefore limiting scalability to large-scale unposed real-world data.

Self-Supervised Pose-Free Methods completely eliminate the reliance on ground-truth poses during training by enabling rendering losses at novel viewpoints using estimated poses, as shown in Fig. 1 (c). For instance, Nope-NeRF [34], CF-3DGS [35], and FlowCam [36] reconstruct 3D scenes and estimate camera poses incrementally by re-rendering dense video sequences. However, they are limited to continuous video frames and do not generalize well to sparse views. Recent self-supervised pose-free methods, such as PF3plat [20] and SelfSplat [21], attempt to estimate both input- and target-view poses from sparse views. PF3plat relies on off-the-shelf feature descriptors [37] with RANSAC-based initialization, yielding a pipeline that is inefficient and not end-to-end trainable, thereby limiting representational capacity. In contrast, SelfSplat employs cross-view U-Nets [38], [39] for pose prediction, but its performance remains weak particularly under large viewpoint changes due to the lack of geometric priors. Beyond the limitations in pose estimation, both methods separate pose prediction and Gaussian reconstruction into distinct modules, resulting in unshared features, weaker geometric alignment, and higher computational overhead. Moreover, both follow a local-to-global strategy: per-pixel depth is first predicted for each view, then lifted into world coordinates using the estimated poses. Pose errors at this stage can directly corrupt the lifted 3D points, degrading reconstruction and amplifying instability through feedback. Consequently, these approaches suffer from unstable training and exhibit a large performance gap compared to state-of-the-art methods.

SPFSplat [22], our previous approach, also adopts a self-supervised, pose-free paradigm. This is accomplished by jointly optimizing 3D Gaussians and camera poses through a shared backbone in a canonical space, guided by both image rendering and reprojection losses. The unified backbone ensures that pose estimation is informed by the same scene geometry that drives Gaussian prediction, thereby promoting geometric consistency and improving training stability. Building upon SPFSplat, we introduce masked attention to reduce computational cost and alleviate potential pose misalignment, enhance pose estimation through learnable pose tokens, and further incorporate a multi-view dropout strategy, which together lead to improved overall performance and generalization across multiple views.

B. Structure-from-Motion (SfM)

Structure-from-Motion (SfM) [40], [41] is a core problem in computer vision that jointly estimates camera parameters and reconstructs sparse 3D maps from image collections. Classical SfM pipelines typically involve local feature detection and matching [42]-[44], geometric verification via epipolar geometry or homographies with RANSAC [45], triangulation [46] to recover 3D points, and bundle adjustment [47] to refine poses and structure. Recent advances have incorporated learning-based components into SfM, including robust feature descriptors [48]-[51], improved image matching [37], [52], detector-free matching [53], and neural bundle adjustment [19], [54]. However, the sequential design of SfM pipelines remains prone to error propagation. To overcome this, fully differentiable pipelines have been introduced [23], [24], [29], [55]–[57]. For example, VGGSfM [55] enables endto-end sparse reconstruction, while DUSt3R [29], following the architecuture of CroCo [58], performs dense 3D reconstruction without camera parameters. MASt3R [23] further enhances feature matching and local representations but, like DUSt3R, remains constrained by pairwise architectures and costly global optimization, which often fail in multi-view settings. Extensions such as MV-DUSt3R+ [56] and Fast3R [57] address multi-view reconstruction, while FLARE [59] employs cascaded learning with pose as the central bridge. Recently, VGGT [24] introduces a feed-forward transformer that jointly infers camera parameters, depth, point maps, and 3D tracks, achieving state-of-the-art results.

Similar to these SfM methods, our method jointly predicts 3D points and poses, with rendering and reprojection losses acting as a differentiable form of bundle adjustment to refine both geometry and poses. Unlike prior work, it requires no ground-truth geometric priors during training. In addition, the training paradigm is naturally compatible with advanced reconstruction backbones such as MASt3R [23] and VGGT [24], giving rise to two variants: SPFSplatV2 and SPFSplatV2-L.

III. METHOD

We aim to learn a feed-forward network that reconstructs 3D Gaussians from unposed images while simultaneously estimating the camera poses. During training, the 3D Gaussians are optimized by rendering photorealistic images from the estimated poses at target views, thereby eliminating the need for ground-truth poses.

A. Problem Formulation

Consider N context images $\{I^v\}_{v=1}^N$ as input. During training, additional M target images $\{I^v\}_{v=N+1}^{N+M}$ are provided, resulting in a total of V=N+M views.

3D Gaussian Reconstruction: Following [7], [8], we predict 3D Gaussians from context images in a canonical 3D space where the first input view I^1 serves as the global coordinate frame. The reconstruction network is formulated as:

$$f_{\boldsymbol{\theta}}: \{\boldsymbol{I}^v\}_{v=1}^N \mapsto \{\boldsymbol{\mathcal{G}}^{v \to 1}\}_{v=1}^N, \tag{1}$$

where $\mathcal{G}^{v \to 1} = \{(\boldsymbol{\mu}_j^{v \to 1}, \boldsymbol{r}_j^{v \to 1}, \boldsymbol{s}_j^v, \boldsymbol{c}_j^{v \to 1}, \alpha_j^v)\}_{j=1,\dots,H \times W}$ represents the pixel-aligned Gaussians for \boldsymbol{I}^v , represented in the coordinate frame of \boldsymbol{I}^1 . Each Gaussian is parameterized by center $\boldsymbol{\mu} \in \mathbb{R}^3$, rotation quaternion $\boldsymbol{r} \in \mathbb{R}^4$, scale $\boldsymbol{s} \in \mathbb{R}^3$, opacity $\alpha \in \mathbb{R}$, and spherical harmonics (SH) $\boldsymbol{c} \in \mathbb{R}^k$ with k degrees of freedom.

Pose Estimation: We introduce a pose network f_{ϕ} to estimate the relative transformation from each view \boldsymbol{I}^v to the reference view \boldsymbol{I}^1 , which is denoted as $\boldsymbol{P}^{v \to 1} = [\boldsymbol{R}^{v \to 1} | \boldsymbol{T}^{v \to 1}]$, where $\boldsymbol{R}^{v \to 1} \in \mathbb{R}^{3 \times 3}$ represents the rotation matrix, and $\boldsymbol{T}^{v \to 1} \in \mathbb{R}^{3 \times 1}$ represents the translation vector. This can be formulated as:

$$\mathbf{P}^{v\to 1} = f_{\phi}(\mathbf{I}^v, \dots, \mathbf{I}^1), v \in [1, \dots, V].$$
 (2)

Novel View Synthesis: Novel views are then rendered using the estimated target poses and reconstructed Gaussians:

$$\hat{I}^t = \mathcal{R}(P^{t \to 1}, \{\mathcal{G}^{v \to 1}\}_{v=1}^N), \quad t \in [N+1, \dots, V].$$
 (3)

B. Architecture

Following state-of-the-art large reconstruction models such as MASt3R [23] and VGGT [24], our framework consists of three main components: an encoder, a decoder, and task-specific prediction heads, as illustrated in Fig. 2. Both the encoder and decoder follow Vision Transformer (ViT) architectures [60]. As shown in Fig. 3, we develop two model variants: SPFSplatV2, which adopts the MASt3R-style architecture, and SPFSplatV2-L, which follows the VGGT design. In the following, we introduce both variants in detail.

Encoder: The RGB image I^v for each input view v is first patchified and flattened into a sequence of image tokens. These tokens are independently processed by a shared-weight ViT encoder, which extracts view-specific feature representations $F^v \in \mathbb{R}^{L \times C}$, where L denotes the number of tokens. This is formulated as follows:

$$\mathbf{F}^v = \text{Encoder}(\mathbf{I}^v), v \in [1, \dots, V].$$
 (4)

Learnable Pose Token: The earlier SPFSplat [22] encodes camera poses by applying global average pooling to decoded

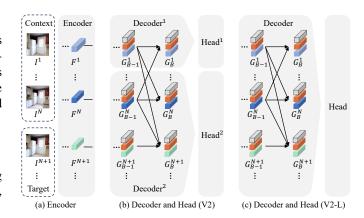


Fig. 3. Architecture comparison of SPFSplatV2 and SPFSplatV2-L. SPFSplatV2 (a + b) uses asymmetrical decoders and heads to distinguish the reference view I^1 from other views, whereas SPFSplatV2-L (a + c) employs a unified decoder and head for all views.

image tokens, enforcing uniform feature aggregation and thereby diluting critical geometric cues. In contrast, SPFS-platV2 introduces a learnable pose token $\boldsymbol{g} \in \mathbb{R}^{1 \times C}$, which is replicated for each view $v \in [1, V]$ as \boldsymbol{g}^v . Unlike SPFS-platV2 which uses an asymmetrical decoder to distinguish the reference frame from the other views, SPFS-platV2-L introduces two separate learnable pose tokens $\bar{\boldsymbol{g}}$ and $\bar{\boldsymbol{g}} \in \mathbb{R}^{1 \times C}$, following VGGT. Specifically, $\bar{\boldsymbol{g}}$ is assigned to the reference frame $(\boldsymbol{g}^1 := \bar{\boldsymbol{g}})$, while all other views share $\bar{\boldsymbol{g}}$ $(\boldsymbol{g}^v := \bar{\boldsymbol{g}}, v \in [2, \ldots, V])$. The pose tokens are concatenated with the encoder tokens, yielding $\boldsymbol{F}^v := [\boldsymbol{g}^v, \boldsymbol{F}^v]$. In the decoding stage, the learnable pose tokens can selectively attend to the most informative features, enabling more accurate pose estimation.

Intrinsics Embedding: Following [7], we encode the camera intrinsics of each view into a token \mathbf{k}^v via a linear layer and concatenate it with the pose token and encoder tokens, forming the decoder input $\mathbf{F}^v := [\mathbf{k}^v, \mathbf{g}^v, \mathbf{F}^v]$. This explicitly injects calibration information, helping to resolve scale ambiguity and improve alignment of predicted poses and 3D Gaussians, particularly under large focal length variations. Importantly, the intrinsic token is optional, and SPFSplatV2 maintains strong performance without it (Sec. IV-D), underscoring the robustness and flexibility of the design.

Masked Multi-view Decoder: To effectively aggregate information across multiple views, we adopt a ViT-based decoder with cross-view attention, enabling joint reasoning over token representations across input views and facilitating cross-view information exchange to capture spatial relationships and the global 3D scene geometry.

During training, since target views are also provided as input, the original SPFSplat [22] adopts a dual-branch design to avoid information leakage from target views into the Gaussian reconstruction of context views. One branch processes only context images for Gaussian prediction, while the other takes both context and target images for pose estimation. However, this design introduces two drawbacks: (i) higher computational cost in cross-attention caused by two forward passes during training, as shown in Fig. 4 (a), and (ii) redundant pose predictions for context views, since each branch produces its own estimates, potentially causing misalignment.

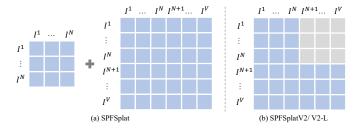


Fig. 4. Comparison of cross-attention in (a) SPFSplat and (b) SPFSplatV2/V2-L. I^1,\cdots,I^N denote context images, while I^{N+1},\cdots,I^V denote target images. SPFSplat relies on dual input branches (context-only and context+target) to block target leakage into context reconstruction, leading to higher cross-attention cost. SPFSplatV2 replaces this with masked attention, enforcing the same separation within a single branch while reducing computation by removing redundant context-target interactions.

To address these issues, we introduce a *masked attention* mechanism as shown in Fig. 3 and Fig. 4 (b). In this approach, context and target images are jointly processed in a single forward pass, with cross-attention selectively masked to control information flow. Specifically, context tokens attend only to context tokens, ensuring Gaussian reconstruction remains independent of target-view information. Meanwhile, target tokens attend to both context and target tokens, allowing the network to leverage global cues for accurate pose estimation.

For SPFSplatV2, we extend MASt3R's pairwise asymmetric decoder to a multi-view setting, which scales efficiently with the number of views while avoiding excessive memory overhead, following similar implementations in [7], [56]. Each decoder block first performs intra-view self-attention, followed by masked cross-attention. Tokens from the first (reference) view are processed with DecoderBlock¹, while tokens from the remaining views use DecoderBlock². The two decoders share the same architecture but maintain independent weights. Formally, the masked decoder block is defined as:

$$\boldsymbol{G}_{i}^{v} = \begin{cases} \text{DecoderBlock}_{i}^{1}(\boldsymbol{G}_{i-1}^{v}, \boldsymbol{G}_{i-1}^{1:K}), & v = 1, \\ \text{DecoderBlock}_{i}^{2}(\boldsymbol{G}_{i-1}^{v}, \boldsymbol{G}_{i-1}^{1:K}), & v \in [2, \dots, V], \end{cases}$$
(5)

for i = 1, ..., B, where B is the number of decoder blocks and $G_0^v = F^v$ are the initial tokens for view v. Here, K = N for context views $(v \in [1, ..., N])$, and K = V for target views $(v \in [N+1, ..., V])$.

For SPFSplatV2-L, we adopt the VGGT architecture, which alternates between intra-frame and inter-frame attention. Different from MASt3R's asymmetric design, the decoder here is unified across all views, which can be expressed as:

$$\boldsymbol{G}_{i}^{v} = \text{DecoderBlock}_{i} \left(\boldsymbol{G}_{i-1}^{v}, \boldsymbol{G}_{i-1}^{1:K} \right), \quad v \in [1, \dots, V], \quad (6)$$

for i = 1, ..., B, where B is the number of decoder blocks, and K follows the same definition as in SPFSplatV2.

Overall, the masked multi-view decoder achieves three key advantages: (i) it preserves generalization to novel viewpoints by strictly preventing target-specific information from contaminating the Gaussian representation, (ii) it significantly reduces computational overhead by avoiding redundant forward passes, and (iii) it eliminates inconsistent context pose estimates, thereby improving geometric alignment. Together,

these improvements lead to more efficient and stable training, as well as more accurate reconstructions.

Gaussian Prediction Heads: Following [7], [8], we employ two DPT-based heads [61] to infer Gaussian parameters. The first head processes decoder tokens of context views and predicts 3D coordinates for each pixel, defining Gaussian centers. The second head estimates rotation, scale, opacity, and SH coefficients for each Gaussian primitive.

For SPFSplatV2, the Gaussian center head extends MASt3R's pairwise asymmetric pointmap head to a multi-view setting by assigning decoder tokens from the first view to the reference head PointHead¹, and tokens from all remaining views to the non-reference head PointHead². The Gaussian parameter head follows the same structure as the Gaussian center head. As proposed in [5]–[7], we incorporate high-resolution skip connections by feeding the original context images into the Gaussian parameter heads, preserving fine-grained spatial details. These heads can be formulated as:

$$\boldsymbol{\mu}^{v \to 1} = \begin{cases} \text{PointHead}^1 \left(\{ \boldsymbol{G}_i^v \}_{i=0}^B \right), & v = 1, \\ \text{PointHead}^2 \left(\{ \boldsymbol{G}_i^v \}_{i=0}^B \right), & v \in [2, \dots, N], \end{cases}$$
(7)

$$\overline{\boldsymbol{\mathcal{G}}}^{v \to 1} = \begin{cases} \operatorname{GSHead}^{1}(\{\boldsymbol{G}_{i}^{v}\}_{i=0}^{B}, \boldsymbol{I}^{v}), & v = 1, \\ \operatorname{GSHead}^{2}(\{\boldsymbol{G}_{i}^{v}\}_{i=0}^{B}, \boldsymbol{I}^{v}), & v \in [2, \dots, N], \end{cases}$$
(8)

where $\{G_i^v\}_{i=0}^B$ denotes the set of decoder tokens taken from different blocks, $\boldsymbol{\mu}^{v \to 1}$ denotes Gaussian centers, and $\overline{\boldsymbol{\mathcal{G}}}^{v \to 1} = \{(\boldsymbol{r}_j^{v \to 1}, \boldsymbol{c}_j^{v \to 1}, \alpha_j^v, \boldsymbol{s}_j^v)\}$ represents rotation, scale, opacity, and SH coefficients for each Gaussian primitive.

For SPFSplatV2-L, we adopt the VGGT design for the pointmap head, which serves as both the Gaussian center head and the Gaussian parameter head. Similar to SPFSplatV2, the Gaussian parameter head additionally also incorporates the original context images as an auxiliary input. Unlike the asymmetric design in SPFSplatV2, the Gaussian prediction heads in SPFSplatV2-L are unified across all views:

$$\boldsymbol{\mu}^{v \to 1} = \text{PointHead}(\{\boldsymbol{G}_i^v\}_{i=0}^B), \quad v \in [1, \dots, N]$$
 (9)

$$\overline{\mathcal{G}}^{v \to 1} = \text{GSHead}(\{G_i^v\}_{i=0}^B, I^v), \quad v \in [1, \dots, N]. \quad (10)$$

Pose Head: After decoding, the attended pose tokens $\hat{g^v}$ are fed into the pose head and further processed by a 3-layer MLP to predict the camera pose as a 10-dimensional representation [62]. The predicted pose representation is decomposed into translation and rotation for each view. The translation is represented using four homogeneous coordinates [62], while the rotation is encoded in a 6D format, capturing two unnormalized coordinate axes. These axes are normalized and combined via a cross-product operation to construct a full rotation matrix [63]. To compute the relative pose with respect to the reference view, the 10D pose representation is converted into a homogeneous transformation matrix $P^{v\to 1} \in \mathbb{R}^{4\times 4}$. Following MASt3R, we make the pose head asymmetrical:

$$\mathbf{P}^{v\to 1} = \begin{cases} \text{PoseHead}^{1}(\hat{\mathbf{g}^{v}}), & v = 1, \\ \text{PoseHead}^{2}(\hat{\mathbf{g}^{v}}), & v \in [2, ..., V], \end{cases}$$
(11)

where $P^{v \to 1}$ is the estimated relative pose from I^v to I^1 .

For SPFSplatV2-L, the pose head follows the original VGGT design: the refined pose tokens \hat{g}^v are subsequently processed by four additional self-attention layers and a linear projection to predict the camera parameters.

$$\mathbf{P}^{v\to 1} = \text{PoseHead}(\hat{\mathbf{g}^v}), v \in [1, ..., V]. \tag{12}$$

We normalize the camera poses by assigning the first input view the canonical pose $[\mathbf{U}|\mathbf{0}]$, where \mathbf{U} represents the identity matrix, and $\mathbf{0}$ denotes the zero translation vector.

C. Loss Functions

Image Rendering Loss: Our model is trained using ground-truth target RGB images as supervision. The training loss is formulated as a weighted combination of the L_2 loss and the LPIPS loss [64], formulated as:

$$\mathcal{L}_{\text{render}} = \|\boldsymbol{I}^t - \hat{\boldsymbol{I}}^t\|_2 + \gamma \text{LPIPS}(\boldsymbol{I}^t, \hat{\boldsymbol{I}}^t), \tag{13}$$

where \boldsymbol{I}^t and $\hat{\boldsymbol{I}}^t$ denote the ground-truth and rendered target images for $t \in [N+1,V]$, and γ is a weighting factor that balances pixel-level accuracy and perceptual similarity.

Reprojection Loss: Existing approaches enforce pixel-aligned Gaussian prediction by constraining Gaussian locations along the input viewing rays [5], [6], [9], [11], [20], [21]. Meanwhile, canonical-space-based methods [7], [8] rely on ground-truth camera poses to guide the canonical 3D points (Gaussian centers). Both strategies ensure alignment between each pixel and its corresponding 3D point. However, since our model learns 3D Gaussian centers in a canonical space without known camera poses, the network lacks explicit geometric constraints to enforce pixel-aligned Gaussian representation.

A naive solution is to include context views in the image rendering loss (Eq. 13) by synthesizing images from them and computing the loss against their ground-truth counterparts. However, this leads to unstable training due to overfitting. Specifically, the network prioritizes improving the rendering quality of the first context view, as the 3D Gaussian space is defined in its camera coordinate, making its rendering independent of the learnable poses. Since the Gaussians from this view already captures sufficient scene information, the model suppresses the contribution of other context views by shifting their Gaussian centers away and adjusting camera poses, ultimately causing training collapse.

To address this issue, we instead employ a pixel-wise reprojection loss to jointly optimize 3D points and camera poses [12], [65]. Unlike purely image-based supervision, this reprojection loss enforces explicit geometric constraints, thereby reducing overfitting to the context views. Concretely, for each pixel \mathbf{p}_j^v in view $v \in [1, N]$, we project the corresponding 3D Gaussian center $\boldsymbol{\mu}_j^{v \to 1}$ from the canonical coordinate frame into the 2D pixel space using the estimated pose of view v, and minimize the reprojection error:

$$\mathcal{L}_{\text{reproj}} = \sum_{v=1}^{N} \sum_{j=1}^{H \times W} \left| \mathbf{p}_{j}^{v} - \pi(\mathbf{K}^{v}, \mathbf{P}^{v \to 1}, \boldsymbol{\mu}_{j}^{v \to 1}) \right|, \quad (14)$$

where π denotes the camera projection function, K^v the camera intrinsics of view v, and $P^{v\to 1}$ the relative pose from view v to the canonical frame.

Different from SPFSplat, which applies reprojection loss to both context-only and context-with-target branches, SPF-SplatV2 predicts a single set of context poses, avoiding redundant supervision and potential pose misalignment. This streamlined design leverages reprojection loss to enable more stable training and efficient optimization of pixel-aligned 3D Gaussians, without requiring ground-truth camera poses.

D. Multi-View Dropout

Unlike SPFSplat, which trains separate models for different numbers of context views, we use a single unified model. To improve generalization, we introduce a multi-view dropout strategy: for more than two context views, the leftmost and rightmost views are retained while intermediate views are randomly dropped during training. This encourages the network to handle flexible input configurations, improves robustness to varying numbers and spatial distributions of views at test time, and implicitly regularizes feature aggregation, leading to more stable training and higher-quality reconstructions.

IV. EXPERIMENTS

We report evaluation results on novel view synthesis quality, cross-dataset generalization, and relative pose estimation across several datasets. In addition, we conduct comprehensive ablation studies to analyze the effectiveness of our method.

A. Experimental Settings

Dataset: We train and evaluate our method RealEstate10K (RE10K) [66], which contains large-scale real estate videos from YouTube, and ACID [67], a dataset of nature scenes captured by aerial drones. Camera poses for both datasets are obtained via SfM, and we follow the official train-test splits used in prior work [5]-[7]. Following [7], evaluations on RE10K and ACID are conducted under varying camera overlaps, where input pairs are grouped by overlap ratios: small (0.05%-0.3%), medium (0.3%-0.55%), and large (0.55%–0.8%), determined using a pretrained dense matcher [68]. To analyze the impact of training scale, we also use DL3DV [69], an outdoor dataset with 10K videos and diverse camera motions beyond RE10K. For cross-dataset generalization, we evaluate on ACID, the object-centric DTU dataset [70], DL3DV, and ScanNet++ [71], which contains indoor scenes with camera trajectories distinct from RE10K.

Baselines: For novel view synthesis, we compare to three groups of baselines: pose-required methods (pixelSplat [5], MVSplat [6]), supervised pose-free methods (CoPoNeRF [17], Splatt3R [8], NoPoSplat [7]), and self-supervised pose-free methods (PF3plat [20], SelfSplat [21], SPFSplat [22]). For camera pose estimation, we compare against SfM-based methods (SuperPoint [48] + SuperGlue [52], DUSt3R [29], MASt3R [23], VGGT [24]) and splatting-based methods (No-PoSplat, SelfSplat, PF3plat, SPFSplat).

Evaluation Protocol: For novel view synthesis, we adopt standard metrics: pixel-level PSNR, patch-level SSIM [72], and feature-level LPIPS [64]. For pose estimation, following prior works [7], [52], we report the area under the cumulative

TABLE I

PERFORMANCE COMPARISON OF NOVEL VIEW SYNTHESIS ON RE10K [66] WITH DIFFERENT IMAGE OVERLAP. OUR METHOD OUTPERFORMS
STATE-OF-THE-ART APPROACHES. THE BEST AND SECOND-BEST RESULTS ARE HIGHLIGHTED, AND * DENOTES EVALUATION WITH POSE ALIGNMENT.

Method		Small			Medium			Large			Average	
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Pose-Required												
pixelSplat	20.277	0.719	0.265	23.726	0.811	0.180	27.152	0.880	0.121	23.859	0.808	0.184
MVSplat	20.371	0.725	0.250	23.808	0.814	0.172	27.466	0.885	0.115	24.012	0.812	0.175
Supervised Pose-I	ree											
CoPoNeRF	17.393	0.585	0.462	18.813	0.616	0.392	20.464	0.652	0.318	18.938	0.619	0.388
Splatt3R	17.789	0.582	0.375	18.828	0.607	0.330	19.243	0.593	0.317	18.688	0.337	0.596
NoPoSplat*	22.514	0.784	0.210	24.899	0.839	0.160	27.411	0.883	0.119	25.033	0.838	0.160
Self-Supervised Po	ose-Free											
SelfSplat	14.828	0.543	0.469	18.857	0.679	0.328	23.338	0.798	0.208	19.152	0.680	0.328
PF3plat	18.358	0.668	0.298	20.953	0.741	0.231	23.491	0.795	0.179	21.042	0.739	0.233
SPFSplat	22.897	0.792	0.201	25.334	0.847	0.153	27.947	0.894	0.110	25.484	0.847	0.153
SPFSplat*	23.178	0.796	0.200	25.695	0.853	0.151	28.377	0.899	0.111	25.845	0.852	0.152
SPFSplatV2	23.123	0.800	0.195	25.542	0.853	0.149	28.143	0.897	0.110	25.693	0.853	0.149
SPFSplatV2*	23.456	0.806	0.193	26.030	0.862	0.145	28.682	0.905	0.107	26.157	0.861	0.146
SPFSplatV2-L	23.138	0.804	0.184	25.518	0.856	0.136	28.081	0.899	0.099	25.668	0.855	0.137
SPFSplatV2-L*	23.329	0.804	0.183	<u>25.863</u>	<u>0.861</u>	0.134	<u>28.456</u>	0.903	$\overline{0.098}$	<u>25.983</u>	0.859	0.136

TABLE II
PERFORMANCE COMPARISON OF NOVEL VIEW SYNTHESIS ON ACID [67]. THE BEST AND SECOND BEST RESULTS ARE HIGHLIGHTED.

Method		Small			Medium			Large			Average	
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Pose-Required												
pixelSplat	22.088	0.655	0.284	25.525	0.777	0.197	28.527	0.854	0.139	25.889	0.780	0.194
MVSplat	21.412	0.640	0.290	25.150	0.772	0.198	28.457	0.854	0.137	25.561	0.775	0.195
Supervised Pose-F	ree											
CoPoNeRF	18.651	0.551	0.485	20.654	0.595	0.418	22.654	0.652	0.343	20.950	0.606	0.406
Splatt3R	17.419	0.501	0.434	18.257	0.514	0.405	18.134	0.508	0.395	18.060	0.510	0.407
NoPoSplat*	23.087	0.685	0.258	25.624	0.777	0.193	28.043	0.841	0.144	25.961	0.781	0.189
Self-Supervised Po	se-Free											
SelfSplat	18.301	0.568	0.408	21.375	0.676	0.314	25.219	0.792	0.214	22.089	0.694	0.298
PF3plat	18.112	0.537	0.376	20.732	0.615	0.307	23.607	0.710	0.228	21.206	0.632	0.293
SPFSplat	22.667	0.665	0.262	25.620	0.773	0.192	28.607	0.856	0.136	26.070	0.781	0.186
SPFSplat*	23.676	0.708	0.243	26.351	0.801	0.182	29.170	0.870	0.131	26.796	0.807	0.176
SPFSplatV2	22.944	0.679	0.255	25.849	0.784	0.187	28.766	0.862	0.133	26.284	0.791	0.182
SPFSplatV2*	23.635	0.700	0.247	26.356	0.798	0.182	29.223	0.871	0.129	26.809	0.804	0.176
SPFSplatV2-L	23.640	0.706	0.225	26.272	0.801	0.166	28.938	0.868	0.120	26.674	0.806	0.162
SPFSplatV2-L*	23.937	0.710	0.224	26.489	0.803	0.165	29.188	0.871	0.118	26.917	0.809	0.160

pose error curve (AUC) at thresholds of 5° , 10° , and 20° , where the pose error is defined as the maximum of the angular errors in rotation and translation.

During evaluation of novel view synthesis, target images are typically rendered with ground-truth poses [5], [6], [8], [17]. An alternative is to render using estimated target poses, as in PF3plat [20] and SelfSplat [21]. NoPoSplat [7] instead adopts an evaluation-time pose alignment (EPA) strategy, which optimizes the target pose during evaluation while keeping the reconstructed Gaussians fixed, so that the rendered image best matches the ground truth. This alignment decouples rendering quality from pose estimation accuracy, enabling direct assessment of Gaussian reconstruction. In contrast, rendering with estimated poses jointly evaluates reconstruction fidelity and the consistency between estimated poses and the learned Gaussians. Unless otherwise noted, we render with estimated poses for comprehensive evaluation and additionally report results with pose alignment for fair comparison to NoPoSplat.

B. Implementation Details.

Our method is implemented in PyTorch and leverages a CUDA-based 3DGS renderer with gradient support for camera poses. All models are trained on a single NVIDIA A100 GPU. Each training sample corresponds to a scene with context and target views, with the frame distance between context views gradually increased during training. The initial learning rate is set to 1×10^{-5} for the backbone and 1×10^{-4} for all other parameters, and LPIPS and reprojection losses are weighted 0.05 and 0.001, respectively. For SPFSplatV2, the encoder adopts a ViT-Large architecture with a patch size of 16, while the decoder is based on ViT-Base. The encoder, decoder, and Gaussian center head are initialized from pretrained MASt3R [23], while the pose head is initialized to approximate the identity rotation matrix for stable convergence. For SPFSplatV2-L, the encoder is a ViT-Large from DINOv2 [73] with a patch size of 14. The encoder, decoder, pose head, and Gaussian center head are initialized



Fig. 5. Qualitative comparison on RE10K (top three rows) and ACID (bottom three rows). Our method 1) better handles extreme viewpoint changes and minimal input overlap (e.g., Row 1 and Row 2), 2) preserves finer details (e.g., Row 3) and more accurate geometric structure (e.g., Row 4 and Row 5), and 3) reduces misaligned blending artifacts and ghosting effect (e.g. Row 5 and Row 6).

from pretrained VGGT [24] weights. All remaining layers are randomly initialized. Training is performed at a resolution of 256×256 and 224×224 for V2 and V2-L, respectively.

C. Results

Novel View Synthesis: Quantitative results on RE10K and ACID are reported in Tab. I and Tab. II. We make the following observations: 1) Despite being trained without ground-truth poses, SPFSplatV2 and SPFSplatV2-L consistently outperform state-of-the-art methods. Notably, both variants outperform our earlier SPFSplat baseline in most cases across different image overlap settings, with or without pose alignment, underscoring the effectiveness of the improved architecture. 2) While evaluation pose alignment generally improves performance, SPFSplatV2 without alignment still outperforms NoPoSplat with alignment, indicating that the jointly optimized poses are well aligned with the reconstructed Gaussians. 3) Between our variants, SPFSplatV2 achieves slightly higher PSNR on RE10K, while SPFSplatV2-L attains better LPIPS scores. On ACID, SPFSplatV2-L delivers the strongest overall results, likely benefiting from VGGT's superior multi-view reconstruction capabilities and feature representations.

Qualitative comparisons in Fig. 5 further demonstrate that our models reduce misalignment and recover more accurate

geometry than baselines, even in challenging scenarios such as minimal input overlap or extreme viewpoint changes. Specifically, SPFSplatV2 improves structural accuracy and visual clarity compared to the original SPFSplat, while SPFSplatV2-L produces the highest overall rendering quality, capturing fine geometric details and textures more faithfully.

Cross-Dataset Generalization: To evaluate zero-shot generalization, we train on RE10K (indoor scenes) and test on ACID (outdoor), DTU (object-centric), DL3DV (outdoor), and ScanNet++ (indoor). As shown in Tab. III, both SPFSplatV2 variants generalize robustly across these diverse domains, consistently outperforming prior approaches. These datasets exhibit substantially different camera motions and scene types compared to RE10K, highlighting the strong out-of-domain generalization capability of our models, even under minimal image overlaps. Notably, in the RE10K→ACID setting, SPFSplatV2 surpasses NoPoSplat and SPFSplat trained directly on ACID (Tab. II) when evaluated with pose alignment. SPFSplatV2-L consistently outperforms SPFSplatV2 both with and without pose alignment.

Qualitative results in Fig. 6 show that both variants produce sharper and more geometrically accurate reconstructions than prior methods, with SPFSplatV2-L achieving the highest visual quality. These results demonstrate that, even without ground-truth poses, our framework effectively aligns

TABLE III

PERFORMANCE COMPARISON OF CROSS-DATASET GENERALIZATION. ALL METHODS ARE TRAINED ON RE10K AND EVALUATED IN A ZERO-SHOT
SETTING ON ACID, DTU, DL3DV AND SCANNET++. OUR METHOD DEMONSTRATES SUPERIOR GENERALIZATION COMPARED TO STATE-OF-THE-ART
APPROACHES. * DENOTES EVALUATION WITH POSE ALIGNMENT.

Method		ACID			DTU			DL3DV			ScanNet++	+
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Pose-Required												
pixelSplat	25.477	0.770	0.207	15.067	0.539	0.341	18.688	0.582	0.354	18.422	0.720	0.278
MVSplat	25.525	0.773	0.199	14.542	0.537	0.324	17.786	0.545	0.357	17.138	0.687	0.297
Supervised Pose-F	ree											
NoPoSplat*	25.764	0.776	0.199	17.899	0.629	0.279	19.974	0.612	0.305	22.136	0.798	0.232
Self-Supervised Po	se-Free											
SelfSplat	22.204	0.686	0.316	13.249	0.434	0.441	15.047	0.410	0.498	13.277	0.538	0.534
PF3plat	20.726	0.610	0.308	12.972	0.407	0.464	15.773	0.458	0.417	16.471	0.688	0.303
SPFSplat	25.965	0.781	0.190	16.550	0.579	0.270	19.172	0.573	0.315	19.971	0.738	0.265
SPFSplat*	26.697	0.806	0.181	18.297	0.660	0.255	19.494	0.574	0.319	22.312	0.793	0.243
SPFSplatV2	$\overline{26.220}$	0.789	0.185	16.793	0.584	0.265	19.439	0.584	0.304	20.919	0.771	0.243
SPFSplatV2*	26.802	0.805	0.179	18.506	0.663	0.246	19.978	0.607	0.302	22.776	0.812	0.227
SPFSplatV2-L	26.361	$\overline{0.796}$	0.169	17.739	0.653	0.228	19.743	0.613	0.277	21.796	0.811	0.200
SPFSplatV2-L*	26.680	0.802	$\overline{0.166}$	19.316	0.671	0.229	20.108	$\overline{0.615}$	0.279	23.072	0.820	0.199

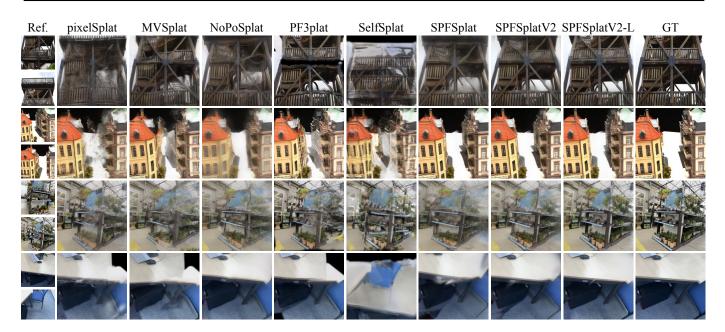


Fig. 6. Qualitative comparison on cross-dataset generalization. All methods are trained on RE10K and evaluated on ACID and DTU, DL3DV, and ScanNet++ (from top to bottom). Both SPFSplatV2 and SPFSplatV2-L yield more geometrically accurate reconstructions than prior methods.

3D Gaussians with predicted camera poses, enabling robust generalization to out-of-distribution scenes.

Relative Pose Estimation. We evaluate relative pose estimation between input image pairs on RE10K, ACID, DL3DV, and ScanNet++, with results in Tab. IV. All splat-based methods are trained on RE10K to evaluate generalization. Since VGGT does not natively support 224 × 224 inputs, we resize and center-crop its input images to 224 × 224 and pad the width to 518, as specified in [24]. SPFSplatV2-L uses 224 × 224 inputs, while all other methods operate at 256 × 256. SuperPoint + SuperGlue derives relative poses from Essential Matrices estimated from feature correspondences. DUSt3R, MASt3R, and NoPoSplat use PnP [40] with RANSAC [45], while PF3plat and VGGT directly predict poses. Our SPFSplat variants support two strategies: (i) direct regression through the

pose head, and (ii) PnP with RANSAC applied to predicted 3D Gaussian centers.

As shown in Tab. IV, both regression and PnP yield similarly strong performance, reflecting consistent alignment between estimated poses and reconstructed 3D points. Despite no geometry priors during training, SPFSplatV2 substantially outperforms MASt3R, its initialization model, and SPFSplatV2-L improves over VGGT in most cases. This demonstrates our framework's ability to jointly optimize camera poses and 3D structure using only image-level supervision. Both SPFSplatV2 variants also significantly surpass the original SPFSplat, primarily due to masked attention improving pose alignment. On RE10K and ACID, our models achieve state-of-the-art results. On DL3DV and ScanNet++, which exhibit challenging camera motions, NoPoSplat achieves better perfor-

TABLE IV
PERFORMANCE COMPARISON OF POSE ESTIMATION IN AUC WITH VARIOUS THRESHOLDS ON RE10K, ACID, DL3DV AND SCANNET++ DATASETS.

Method		RE10K			ACID			DL3DV			ScanNet++	+
	5° ↑	10° ↑	20° ↑	5° ↑	10° ↑	20° ↑	5° ↑	10° ↑	20° ↑	5° ↑	10° ↑	20° ↑
SfM												
SP + SG	0.234	0.406	0.569	0.228	0.363	0.500	0.224	0.372	0.492	0.087	0.151	0.248
DUSt3R	0.336	0.541	0.702	0.118	0.279	0.470	0.275	0.490	0.686	0.109	0.284	0.500
MASt3R	0.281	0.494	0.672	0.138	0.312	0.507	0.332	0.593	0.772	0.139	0.336	0.549
VGGT	0.257	0.474	0.658	0.142	0.304	0.486	0.356	0.609	0.784	0.156	0.311	0.514
Pose-Free View Synthesis	S											
NoPoSplat	0.571	0.727	0.833	0.335	0.496	0.644	0.470	0.646	0.762	0.207	0.403	0.641
PF3plat	0.187	0.398	0.613	0.060	0.165	0.340	0.118	0.281	0.479	0.058	0.204	0.415
SPFSplat	0.617	0.755	0.845	0.364	0.520	0.662	0.283	0.461	0.622	0.098	0.188	0.374
SPFSplat (PnP)	0.613	0.754	0.845	0.355	0.516	0.658	0.279	0.464	0.626	0.120	0.226	0.408
SPFSplatV2	0.638	0.776	0.863	0.387	0.541	0.672	0.369	0.534	0.694	0.144	0.281	0.487
SPFSplatV2 (PnP)	0.641	0.777	0.864	0.374	0.533	0.667	0.375	0.542	0.700	0.111	0.250	0.463
SPFSplatV2-L	0.645	0.780	0.864	0.379	0.539	0.671	0.420	0.582	0.711	0.184	0.400	0.630
SPFSplatV2-L (PnP)	0.657	0.786	0.867	0.375	$\overline{0.535}$	0.668	0.429	0.587	0.716	0.183	0.400	0.627

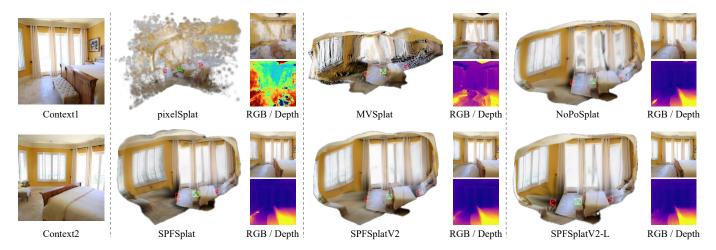


Fig. 7. Comparison of 3D Gaussians and rendered results. Red and green denote context and target camera poses, respectively. Rendered images and depth maps at the target views are shown on the right. Our method produces higher-quality 3D Gaussians and better rendering over baselines.

mance, benefiting from ground-truth pose supervision during training. As shown in Sec. IV-D, this gap can be eliminated by scaling our approach to larger training datasets.

Geometry Reconstruction: As illustrated in Fig. 7, SPFS-platV2 and SPFSplatV2-L produce substantially higher-quality 3D Gaussian primitives than prior methods, even under large viewpoint changes between input pairs. Previous approaches often exhibit distorted structures or ghosting artifacts, whereas our models, trained without ground-truth poses, reconstruct more accurate 3D geometry and yield sharper renderings, reflecting improved Gaussian alignment across views. This improvement arises from the joint optimization of Gaussians and poses, which strengthens geometric consistency. Compared to SPFSplat, SPFSplatV2 achieves more precise structural reconstruction, particularly visible in the left windows, while SPFSplatV2-L further enhances overall Gaussian quality.

Extension to Multiple Views: Our method naturally extends to multiple input views. As shown in Tab. V, novel view synthesis performance consistently improves with more context views. Both SPFSplatV2 and SPFSplatV2-L outperform the NoPoSplat and original SPFSplat, benefiting from enhanced architectures and the multi-view dropout strategy. With denser

inputs, SPFSplatV2-L shows a clearer advantage, as its VGGT backbone, pretrained on multi-view data, provides stronger representations than MASt3R, which is limited to pairwise training. These results show that SPFSplatV2-L can better leverage additional views to enhance geometric consistency and reconstruction fidelity. Overall, the consistent gains with increasing context views highlight the flexibility and scalability of our framework for multi-view scenarios.

Efficiency: We compare the efficiency of our method to other approaches in Tab. VI and Tab. VII.

1) Inference Efficiency: Tab. VI reports parameter size, FLOPs, and runtime during inference, measured for reconstructing 3D Gaussians from two input images on an A6000 GPU. The input resolution is 256×256 , except for SPFSplatV2-L which uses 224×224 . SPFSplatV2 achieves comparable model size, FLOPs, and runtime to NoPoSplat and SPFSplat, while providing substantial speedups of approximately $3.5 \times$, $1.4 \times$, $2.3 \times$, and $27 \times$ over pixelSplat, MVSplat, SelfSplat, and PF3plat, respectively. These gains stem from architectural differences: pixelSplat requires costly cost-volume construction; and both SelfSplat and PF3plat

TABLE V Novel view synthesis with varying input view numbers. For NoPoSplat, only results reported in [7] are shown.

Method	2 Views			3 Views			5 Views			10 Views		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NoPoSplat*	25.033	0.838	0.160	26.619	0.872	0.127	_	_	_	_	_	_
SPFSplat	25.484	0.847	0.153	26.724	0.871	0.128	26.891	0.875	0.122	27.159	0.880	0.115
SPFSplatV2 SPFSplatV2-L	25.693 25.668	$\frac{0.853}{0.855}$	$\frac{0.149}{0.137}$	27.262 27.685	$\frac{0.884}{0.898}$	$\frac{0.120}{0.101}$	27.585 28.141	0.890 0.907	$\frac{0.115}{0.094}$	28.188 28.973	$\frac{0.901}{0.922}$	$\frac{0.106}{0.083}$

TABLE VI Comparison of inference Efficiency on An NVIDIA A6000 GPU.

Methods	Params (B)	Inference FLOPs (T)	Inference Time (s)
pixelSplat	0.119	0.764	0.152
MVSplat	0.012	0.170	0.059
NoPoSplat	0.612	0.405	0.042
SelfSplat	0.081	0.491	0.101
PF3plat	0.394	2.164	1.171
SPFSplat	0.616	0.405	0.044
SPFSplatV2	0.613	0.405	0.043
SPFSplatV2-L	1.223	0.610	0.075

depend on separate pose-estimation modules to lift predicted depth into Gaussians, with PF3plat further incurring heavy local feature matching costs. In contrast, SPFSplatV2 reconstructs Gaussians directly in a canonical space using a feed-forward network, avoiding explicit geometric operations such as cost-volume construction. Compared to SPFSplatV2, SPFSplatV2-L introduces additional computational overhead, as a trade-off for superior reconstruction quality.

2) Training Efficiency: For a fair comparison, we evaluate the training efficiency of SPFSplat variants only against PF3plat and SelfSplat, as self-supervised pose-free methods require both context and target images during training, whereas other methods use only context images. The results are summarized in Tab. VII. For all methods, two images are used as context views and one as the target view. Training time and GPU memory usage are measured on an NVIDIA A100 and averaged per sample. PF3plat requires significantly larger FLOPs, time, and GPU memory during training. Thanks to masked attention, SPFSplatV2 reduces training FLOPs of SPFSplat by 12%, resulting in a 25% speedup and a 13% reduction in memory consumption. In contrast, SPFSplatV2-L incurs higher computational and memory costs, with a 46% increase in FLOPs, 30% longer training time, and 25% higher memory usage compared to SPFSplat, reflecting the trade-off for its superior reconstruction performance.

D. Ablation Analysis

Scaling to Larger Training Data: Since our approach does not rely on ground-truth poses, it scales efficiently to larger datasets with minimal annotation. To evaluate the effect of training data size, we train SPFSplatV2 and SPFSplatV2-L on a combination of RE10K and DL3DV, and assess pose estimation on RE10K and DL3DV (in-domain), as well as ACID and ScanNet++ (out-of-domain). As shown in Tab. VIII, enlarging the training set consistently improves both direct regression and PnP-based pose estimation, driven by the greater

TABLE VII
COMPARISON OF TRAINING EFFICIENCY ON AN NVIDIA A100 GPU.

Methods	Training FLOPs (T)	Training Time (s)	Mem. (GB)
SelfSplat	0.491	0.122	6.602
PF3plat	2.164	0.633	15.043
SPFSplat	0.582	0.110	5.634
SPFSplatV2	0.515	0.082	4.891
SPFSplatV2-L	0.849	0.143	7.044

diversity of camera trajectories and scene appearances introduced by DL3DV. Among the models, SPFSplatV2-L achieves the strongest results across most benchmarks. Compared to Tab. IV, incorporating DL3DV during training enables both variants to surpass all prior methods, including NoPoSplat, across different benchmarks, highlighting the scalability and effectiveness of our framework without relying on any ground-truth pose supervision.

Ablation on Different Components: We conduct an ablation study to assess the contribution of individual components in our framework, as summarized in Tab. IX. Setting (a) corresponds to SPFSplatV2, while setting (c) corresponds to SPFSplat. Compared to (a), setting (b) replaces the learnable pose token with global average pooling over feature maps, leading to a slight performance drop. This highlights the effectiveness of the improved pose estimation enabled by the learnable token. Compared with (c), setting (b) substitutes SPFSplat's two-branch input design with masked attention yields consistent improvements in both novel view synthesis and pose estimation, demonstrating the benefits of masked attention. From (a) to (d), removing intrinsic embeddings in the backbone slightly reduces accuracy, primarily due to increased scale ambiguity in both 3D Gaussian learning and pose estimation. Nevertheless, even without intrinsic embeddings, our method still surpasses NoPoSplat with intrinsic embeddings (Tab. I), achieving improvements of 2.25% in PSNR, 1.07% in SSIM, and 4.38% in LPIPS under pose alignment evaluation. Finally, comparing (a) with (e), removing the reprojection loss while retaining only the image rendering loss on target views leads to a substantial degradation in both novel view synthesis and pose estimation accuracy. This underscores the crucial role of geometric constraints between 3D points and camera poses for accurate reconstruction.

In Tab. X, we evaluate our framework's ability to reconstruct geometry without ground-truth pose supervision by analyzing the effect of incorporating ground-truth poses during training in two settings: (b) rendering novel views using ground-truth poses, as in NoPoSplat, and (c) introducing a pose loss that

TABLE VIII

Pose estimation performance using an augmented training set (RE10K + DL3DV). The improvement percentage is calculated relative to the performance shown in Tab. IV.

Method	RE10K			ACID			DL3DV			ScanNet++		
1/10/11/04	5° ↑	10° ↑	20° ↑	5° ↑	10° ↑	20° ↑	5° ↑	10° ↑	20° ↑	5° ↑	10° ↑	20° ↑
SPFSplatV2	0.652 +2.19%	0.785	0.867	0.390	0.543 +0.37%	0.675 +0.45%	0.560 +51.76%	0.711 +33.15%	0.806 +16.14%	0.251 +74.31%	0.478 +70.11%	0.698 +43.33%
SPFSplatV2 (PnP)	0.652 +1.72%	0.784 +0.90%	0.867 +0.35%	0.383 +2.41%	0.538 +0.94%	0.672 +0.75%	0.559 +49.07%	0.714 +31.73%	0.809 +15.57%	0.288 +159.46%	0.493 +97.20%	0.702 +51.62%
SPFSplatV2-L	0.654	0.788	0.870	0.405	0.558 +3.53%	0.687	0.568 +35,24%	0.713 +22.51%	0.809	0.268	0.473 +18.25%	0.670 +6.35%
SPFSplatV2-L (PnP)	0.668	0.794 +1.02%	0.873 +0.69%	0.404 +7.73%	0.556 +3.93%	0.685 +2.54%	0.568	0.717 +22.15%	0.811 +13.27%	0.261 +42.62%	0.469 +17.25%	0.667

TABLE IX Component ablations on RE10K. NVS * denotes novel view synthesis evaluated with pose alignment.

#	M.	P.	I.	R.		NVS		NVS*			Pose			Pose (PnP)		
			-		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	5° ↑	10° ↑	20° ↑	5° ↑	10° ↑	20° ↑
(a)	/	1	/	1	25.693	0.853	0.149	26.157	0.861	0.146	0.638	0.776	0.863	0.641	0.777	0.864
(b)	/	Х	/	1	25.636	0.851	0.150	26.098	0.859	0.147	0.634	0.772	0.859	0.632	0.770	0.858
(c)	X	Х	/	/	25.484	0.847	0.153	25.845	0.852	0.152	0.617	0.755	0.845	0.613	0.754	0.845
(d)	/	/	Х	/	24.998	0.834	0.157	25.597	0.847	0.153	0.546	0.716	0.829	0.581	0.738	0.841
(e)	✓	1	1	X	19.818	0.651	0.280	22.013	0.751	0.244	0.023	0.144	0.393	0.015	0.064	0.197

[&]quot;M." indicates masked attention. "P." indicates learnable pose token. "I." indicates intrinsics embedding. "R." indicates reprojection loss.

TABLE X
GROUND-TRUTH POSES ABLATIONS ON RE10K.

Method		NVS^*	Pose			
	PSNR↑	SSIM↑	LPIPS↓	5° ↑	10° ↑	20° ↑
(a) SPFSplatV2 (Ours) (b) render with gt pose	25.033	0.861 0.838	0.146 0.160	0.571	0.727	0.863 0.833 0.889
(b) render with gt pose (c) w/ gt pose loss	25.033 25.910	0.838 <u>0.860</u>	0.160 <u>0.150</u>	0.571 0.693		

TABLE XI
COMPARISON OF DIFFERENT INITIALIZATION STRATEGIES.

Method	Initialization	PSNR↑	SSIM↑	LPIPS↓
SPFSplatV2	Random	22.394	0.737	0.230
	DUSt3R	25.800	0.854	0.150
	MASt3R	26.157	0.861	0.146
SPFSplatV2-L	Random	22.226	0.724	0.224
	VGGT	25.983	0.859	0.136

penalizes the discrepancy between predicted and ground-truth poses, while still rendering with predicted poses. The pose loss combines a geodesic loss [74] for rotation and an L_2 loss for translation. Since ground-truth poses are used only during training, the framework remains pose-free at inference.

Relative to setting (b), (a) SPFSplatV2 achieves improvements in both novel view synthesis and pose estimation. This can be attributed to the joint optimization of Gaussians and poses, which encourages better geometric alignment and more consistent feature learning. From (a) to (c), adding a pose loss improves pose accuracy but yields only marginal gains in novel view synthesis, underscoring the model's capacity to reconstruct geometry without explicit pose supervision. These findings also suggest that high-quality novel view synthesis depends on factors beyond pose accuracy, such as occlusion, textureless regions, and extreme viewpoint changes, which may require generative priors or explicit 3D supervision.

Ablation on Initialization: In our main experiments, SPFSplatV2 is initialized with MASt3R weights, while SPFSplatV2-L uses VGGT weights. Tab. XI further analyzes the effect of different initialization strategies, showing that MASt3R slightly outperforms DUSt3R, likely due to

its feature-matching pretraining, which yields stronger local features that benefit both pose estimation and 3D Gaussian reconstruction. For random initialization, we employ a warm-up phase with a point cloud distillation loss from DUSt3R during the first 10,000 steps. This additional supervision is essential, as noted in [7], because training solely with a photometric loss, especially without ground-truth geometric supervision, makes it difficult for the network to learn Gaussians in the canonical space. While random initialization leads to a clear performance drop, the results still demonstrate the model's ability to reconstruct Gaussians. Notably, performance under random initialization remains significantly higher than SelfSplat (Tab. I), which also avoids pretrained 3D priors but relies on CroCoV2 [75] weights for feature extraction.

Evaluation on In-the-Wild Data: We highlight the effectiveness of our model on mobile phone photos using SPFSplatV2 without intrinsic embeddings. The 3D geometry and rendered results in Fig. 8 demonstrate strong out-of-domain generalization, even under large viewpoint changes.

Failure Cases: As shown in Fig. 9, our method can produce blurred outputs or artifacts in occluded or texture-less regions, or under extreme viewpoint changes. Addressing these limi-



Fig. 8. 3D Gaussians from smartphone without intrinsics and rendered image.



Fig. 9. Failure cases of SPFSplatV2. Blurriness and artifacts occur in occluded or texture-less regions and under extreme viewpoint changes.

tations may require stronger generative capabilities or larger training data.

V. LIMITATIONS AND FUTURE WORK

Our method can be trained without ground-truth poses and scales effectively to large datasets, opening the possibility for future work to exploit more diverse data to further improve pose estimation and generalization. Nonetheless, it still benefits from the priors provided by supervised models such as MASt3R and VGGT, as evidenced by the performance drop when training from random initialization. Furthermore, since our approach is not generative, it cannot reconstruct unseen regions with high-fidelity textures. Incorporating generative models is a promising direction to address this limitation.

VI. CONCLUSION

This paper presents SPFSplatV2, a self-supervised pose-free framework for 3D Gaussian splatting from sparse unposed views. By jointly optimizing camera poses and 3D Gaussian primitives through a unified backbone with masked attention, our approach achieves efficient and stable training as well as strong geometric consistency without requiring ground-truth poses. A reprojection loss is also incorporated with the conventional rendering loss to enforce pixel-aligned Gaussians. Extensive experiments on multiple datasets demonstrate that SPFSplatV2 and its larger variant SPFSplatV2-L establish new state-of-the-art results in novel view synthesis, crossdataset generalization, and relative pose estimation, even under challenging conditions of extreme viewpoint change and limited overlap. Importantly, the framework's independence from ground-truth poses underscores its scalability to large and diverse real-world datasets, paving the way for future advances in scalable and generalizable 3D reconstruction.

REFERENCES

- 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.
- [2] 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.
- [3] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su, "Mysnerf: Fast generalizable radiance field reconstruction from multi-view stereo," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 14124–14133.
- [4] H. Xu, A. Chen, Y. Chen, C. Sakaridis, Y. Zhang, M. Pollefeys, A. Geiger, and F. Yu, "Murf: multi-baseline radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20041–20050.
- [5] 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. 19457–19467.
- [6] Y. Chen, H. Xu, C. Zheng, B. Zhuang, M. Pollefeys, A. Geiger, T.-J. Cham, and J. Cai, "Mysplat: Efficient 3d gaussian splatting from sparse multi-view images," in *European Conference on Computer Vision*. Springer, 2024, pp. 370–386.
- [7] B. Ye, S. Liu, H. Xu, X. Li, M. Pollefeys, M.-H. Yang, and S. Peng, "No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=P4o9akekdf
- [8] B. Smart, C. Zheng, I. Laina, and V. A. Prisacariu, "Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs," arXiv preprint arXiv:2408.13912, 2024.
- [9] K. Zhang, S. Bi, H. Tan, Y. Xiangli, N. Zhao, K. Sunkavalli, and Z. Xu, "Gs-lrm: Large reconstruction model for 3d gaussian splatting," in *European Conference on Computer Vision*. Springer, 2024, pp. 1–19.
- [10] J. Tang, Z. Chen, X. Chen, T. Wang, G. Zeng, and Z. Liu, "Lgm: Large multi-view gaussian model for high-resolution 3d content creation," in European Conference on Computer Vision. Springer, 2024, pp. 1–18.
- [11] Y. Xu, Z. Shi, W. Yifan, H. Chen, C. Yang, S. Peng, Y. Shen, and G. Wetzstein, "Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation," in *European Conference on Computer Vision*. Springer, 2024, pp. 1–20.
- [12] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.
- [13] H. Jiang, Z. Jiang, Y. Zhao, and Q. Huang, "LEAP: Liberate sparse-view 3d modeling from camera poses," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=KPmajBxEaF
- [14] P. Wang, H. Tan, S. Bi, Y. Xu, F. Luan, K. Sunkavalli, W. Wang, Z. Xu, and K. Zhang, "PF-LRM: Pose-free large reconstruction model for joint pose and shape prediction," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=noe76eRcPC
- [15] B. R. Nagoor Kani, H.-Y. Lee, S. Tulyakov, and S. Tulsiani, "Upfusion: Novel view diffusion from unposed sparse view observations," in European Conference on Computer Vision (ECCV), 2024.
- [16] M. S. Sajjadi, H. Meyer, E. Pot, U. Bergmann, K. Greff, N. Radwan, S. Vora, M. Lučić, D. Duckworth, A. Dosovitskiy et al., "Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 6229–6238.
- [17] S. Hong, J. Jung, H. Shin, J. Yang, S. Kim, and C. Luo, "Unifying correspondence pose and nerf for generalized pose-free novel view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer* Vision and Pattern Recognition, 2024, pp. 20196–20206.
- [18] Y. Chen and G. H. Lee, "Dbarf: Deep bundle-adjusting generalizable neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 24–34.
- [19] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in *Proceedings of the IEEE/CVF international* conference on computer vision, 2021, pp. 5741–5751.
- [20] S. Hong, J. Jung, H. Shin, J. Han, J. Yang, C. Luo, and S. Kim, "Pf3plat: Pose-free feed-forward 3d gaussian splatting," arXiv preprint arXiv:2410.22128, 2024.

- [21] G. Kang, J. Yoo, J. Park, S. Nam, H. Im, S. Shin, S. Kim, and E. Park, "Selfsplat: Pose-free and 3d prior-free generalizable 3d gaussian splatting," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 22012–22022.
- [22] R. Huang and K. Mikolajczyk, "No pose at all: Self-supervised pose-free 3d gaussian splatting from sparse views," *arXiv preprint arXiv:2508.01171*, 2025.
- [23] V. Leroy, Y. Cabon, and J. Revaud, "Grounding image matching in 3d with mast3r," in *European Conference on Computer Vision*. Springer, 2024, pp. 71–91.
- [24] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, "Vggt: Visual geometry grounded transformer," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 5294–5306.
- [25] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," ACM transactions on graphics (TOG), vol. 41, no. 4, pp. 1–15, 2022.
- [26] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa, "K-planes: Explicit radiance fields in space, time, and appearance," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12479–12488.
- [27] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *European conference on computer vision*. Springer, 2022, pp. 333–350.
- [28] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5501–5510.
- [29] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 697–20 709.
- [30] J. Y. Zhang, D. Ramanan, and S. Tulsiani, "Relpose: Predicting probabilistic relative rotation for single objects in the wild," in *European Conference on Computer Vision*. Springer, 2022, pp. 592–611.
- [31] A. Lin, J. Y. Zhang, D. Ramanan, and S. Tulsiani, "Relpose++: Recovering 6d poses from sparse-view observations," in 2024 International Conference on 3D Vision (3DV). IEEE, 2024, pp. 106–115.
- [32] J. Wang, C. Rupprecht, and D. Novotny, "Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment," in *Proceedings of* the IEEE/CVF International Conference on Computer Vision, 2023, pp. 9773–9783.
- [33] P. Truong, M.-J. Rakotosaona, F. Manhardt, and F. Tombari, "Sparf: Neural radiance fields from sparse and noisy poses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4190–4200.
- [34] W. Bian, Z. Wang, K. Li, J.-W. Bian, and V. A. Prisacariu, "Nope-nerf: Optimising neural radiance field with no pose prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4160–4169.
- [35] Y. Fu, S. Liu, A. Kulkarni, J. Kautz, A. A. Efros, and X. Wang, "Colmapfree 3d gaussian splatting," in *Proceedings of the IEEE/CVF Conference* on Computer Vision and Pattern Recognition, 2024, pp. 20796–20805.
- [36] C. Smith, Y. Du, A. Tewari, and V. Sitzmann, "Flowcam: Training generalizable 3d radiance fields without camera poses via pixel-aligned scene flow," *Advances in Neural Information Processing Systems*, vol. 36, pp. 1476–1488, 2023.
- [37] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "Lightglue: Local feature matching at light speed," in *Proceedings of the IEEE/CVF International* Conference on Computer Vision, 2023, pp. 17 627–17 638.
- [38] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and* computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. Springer, 2015, pp. 234–241.
- [39] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10684–10695.
- [40] R. Hartley and A. Zisserman, Multiple view geometry in computer vision. Cambridge university press, 2003.
- [41] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.
- [42] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International journal of computer vision, vol. 60, pp. 91–110, 2004.

- [43] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," Comput. Vis. Image. Und., vol. 110, no. 3, pp. 346– 359, 2008.
- [44] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proc. IEEE Int. Conf. Comput. Vision.* (ICCV). Ieee, 2011, pp. 2564–2571.
- [45] M. FISCHLER AND, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [46] R. I. Hartley and P. Sturm, "Triangulation," Computer vision and image understanding, vol. 68, no. 2, pp. 146–157, 1997.
- [47] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International workshop on vision algorithms*. Springer, 1999, pp. 298–372.
- [48] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.
- [49] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-Net: A Trainable CNN for Joint Detection and Description of Local Features," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.
- [50] R. Huang, J. Cai, C. Li, Z. Wu, X. Liu, and Z. Chai, "Drkf: Distilled rotated kernel fusion for efficient rotation invariant descriptors in local feature matching," in 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2023, pp. 1885–1892.
- [51] Z. Luo, L. Zhou, X. Bai, H. Chen, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan, "Aslfeat: Learning local features of accurate shape and localization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6589–6598.
- [52] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings* of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 4938–4947.
- [53] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "Loftr: Detectorfree local feature matching with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8922–8931.
- [54] C. Tang and P. Tan, "Ba-net: Dense bundle adjustment network," arXiv preprint arXiv:1806.04807, 2018.
- [55] J. Wang, N. Karaev, C. Rupprecht, and D. Novotny, "Vggsfm: Visual geometry grounded deep structure from motion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 21 686–21 697.
- [56] Z. Tang, Y. Fan, D. Wang, H. Xu, R. Ranjan, A. Schwing, and Z. Yan, "Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 5283–5293.
- [57] J. Yang, A. Sax, K. J. Liang, M. Henaff, H. Tang, A. Cao, J. Chai, F. Meier, and M. Feiszli, "Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 21924–21935.
- [58] P. Weinzaepfel, V. Leroy, T. Lucas, R. Brégier, Y. Cabon, V. Arora, L. Antsfeld, B. Chidlovskii, G. Csurka, and J. Revaud, "Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion," Advances in Neural Information Processing Systems, vol. 35, pp. 3502–3516, 2022.
- [59] S. Zhang, J. Wang, Y. Xu, N. Xue, C. Rupprecht, X. Zhou, Y. Shen, and G. Wetzstein, "Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 21936– 21947.
- [60] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy
- [61] 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. 12179–12188.
- [62] E. Brachmann, T. Cavallari, and V. A. Prisacariu, "Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5044–5053.
- [63] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the*

- *IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5745–5753.
- [64] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 586–595.
- [65] E. Brachmann, J. Wynn, S. Chen, T. Cavallari, Á. Monszpart, D. Turmukhambetov, and V. A. Prisacariu, "Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer," in *European Conference on Computer Vision*. Springer, 2024, pp. 421–440.
- [66] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, "Stereo magnification: learning view synthesis using multiplane images," ACM Transactions on Graphics (TOG), vol. 37, no. 4, pp. 1–12, 2018.
- [67] A. Liu, R. Tucker, V. Jampani, A. Makadia, N. Snavely, and A. Kanazawa, "Infinite nature: Perpetual view generation of natural scenes from a single image," in *Proceedings of the IEEE/CVF Inter*national Conference on Computer Vision, 2021, pp. 14458–14467.
- [68] J. Edstedt, Q. Sun, G. Bökman, M. Wadenbäck, and M. Felsberg, "Roma: Robust dense feature matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19790–19800.
- [69] L. Ling, Y. Sheng, Z. Tu, W. Zhao, C. Xin, K. Wan, L. Yu, Q. Guo, Z. Yu, Y. Lu et al., "Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 22160–22169.
- [70] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs, "Large scale multi-view stereopsis evaluation," in *Proceedings of the IEEE conference* on computer vision and pattern recognition, 2014, pp. 406–413.
- [71] C. Yeshwanth, Y.-C. Liu, M. Nießner, and A. Dai, "Scannet++: A high-fidelity dataset of 3d indoor scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 12–22.
- [72] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [73] 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.
- [74] S. S. M. Salehi, S. Khan, D. Erdogmus, and A. Gholipour, "Real-time deep pose estimation with geodesic loss for image-to-template rigid registration," *IEEE transactions on medical imaging*, vol. 38, no. 2, pp. 470–481, 2018.
- [75] P. Weinzaepfel, T. Lucas, V. Leroy, Y. Cabon, V. Arora, R. Brégier, G. Csurka, L. Antsfeld, B. Chidlovskii, and J. Revaud, "Croco v2: Improved cross-view completion pre-training for stereo matching and optical flow," in *Proceedings of the IEEE/CVF International Conference* on Computer Vision, 2023, pp. 17969–17980.