IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

High-quality Sparse-view Gaussian Splatting without Ground-truth Camera Poses

Lim Chun Her[1], Yingnan Guo[1], Wen Yang[2], Yu Zhang[1,*]

***Abstract*— The existing methods for novel view synthesis depend on dense input images and accurate camera poses, which significantly limits their practical application. We propose a novel framework that enables high-quality sparse-view reconstruction via 3D Gaussian Splatting (3DGS) without knowing camera poses. Our approach leverages MASt3R, a ViT-based multi-view stereo prior, to generate point clouds and coarse camera poses from uncalibrated sparse images. We use the point clouds to initial 3DGS. Additionally, we propose several regularization techniques, including point-rendered LPIPS regularization, geometric regularization (local depth regularization and normal regularization), and semantic regularization to improve the quality of reconstructed scenes and enhance the generalization capability of the model in unseen viewpoint. Due to the inaccuracies in the camera poses output by MASt3R, we optimized the camera poses during both the training and testing phase. Experimental results on the Tanks and Temples and MVImgNet datasets demonstrate that our method outperforms state-of-the-art techniques in novel view synthesis and camera pose estimation under sparse-view settings. Our approach achieves higher fidelity and more photorealistic visual effects.**

I. INTRODUCTION

Recent advancements in implicit 3D scene representations, particularly Neural Radiance Fields (NeRF) [1], have significantly enhanced the fidelity of novel view synthesis and scene reconstruction. Despite their impressive capabilities, NeRF-based methods are hindered by extensive training and inference times, limiting their practicality in real-world applications. To address these limitations, Kerbl et al. [2] introduced 3D Gaussian Splatting, which not only accelerates the training process but also achieves superior reconstruction quality compared to NeRF. However, both NeRF and 3DGS heavily rely on dense image inputs often requiring hundreds of images and accurate camera parameters, typically obtained through Structure-from-Motion (SfM) [3] techniques. This dependency poses significant challenges for deployment in real-world scenarios such as autonomous driving and robot navigation, where acquiring extensive input images and precise camera poses is often impractical [4], [5]. Moreover, under sparse-view conditions, traditional SfM algorithms struggle due to insufficient overlapping regions between images, leading to unreliable feature matching and, consequently, inaccurate camera pose estimations.

In sparse-view settings, 3DGS tends to overfit to the limited input views, resulting in artifacts such as floaters and background collapse. These reconstruction failures not only degrade the visual quality of synthesized views but also compromise the overall coherence and realism of the reconstructed scene, making it challenging to achieve the high fidelity required for specialized domains. Additionally, sparse-view reconstructions often suffer from geometric degradation. With fewer input views, the model struggles to accurately learn the underlying 3D structure, leading to incomplete or incorrect geometric representations. This is particularly pronounced in regions with complex textures or intricate details. Numerous approaches [5], [6], [7], [8] have been proposed to create novel view from sparse input images. However, they generally rely on the assumption that camera poses are known, a condition that is nearly impossible to achieve in real-world scenarios. InstantSplat [9] propose a method for sparse-view reconstruction without knowing camera poses. However, their approach primarily focuses on reconstruction speed and lacks sufficient constraints, often resulting in suboptimal reconstructions in many scenes.

In this paper, we propose a novel approach that enables high-quality sparse-view reconstruction without knowing camera poses. To address the challenge that sparse-view fail to generate initial point clouds and camera extrinsics via SfM, we apply MASt3R [10], a ViT-based [11] multi-view stereo (MVS) prior model. MASt3R generates pixel-aligned point maps and estimates camera poses from uncalibrated images, which we utilize to initialize the 3DGS framework. Recognizing that sparse-view 3D reconstruction is inherently ill-posed [8], we incorporate priors from pretrained deep learning models to provide additional constraints. To improve multi-view consistency, we interpolated the coarse camera poses of the training images output by MASt3R to obtain pseudo camera poses and compute the LPIPS [12] between images rendered using point-based and Gaussian-based methods under these pseudo camera poses, incorporating this metric as an auxiliary loss term.

Furthermore, we employ monocular depth and normal estimation models [13] to constrain the rendered depth and normal maps, addressing geometric degradation in sparse-view reconstructions. We adopt a local depth regularization strategy to more effectively reconstruct fine-grained scene details [6], [14]. Finally, we enhance semantic consistency across different viewpoints by applying semantic regularization. Specifically, we extract semantic feature vectors from both input images and rendered images using the DINO-ViT [15] model and minimize the distance between these vectors in the latent space.

In summary, the main contributions of this work can be

[1]State Key Laboratory of Industrial Control Technology, College of Control Sciience and Engineering, Zhejiang University, Hangzhou, China.
[2]Zhejiang University
* Correspondence: Yu Zhang. Email: zhangyu80@zju.edu.cn

Preprint submitted to IEEE Robotics and Automation Letters as Submission for RA-L only
Received June 17, 2025 07:43:52 Pacific Time

summarized as follows:

- We propose a method capable of synthesizing high-quality novel views under sparse-view settings without knowing camera poses
- We introduce multiple regularization techniques, which effectively mitigate overfitting issues while enhancing fine-grained details in reconstructed scenes.
- Our approach outperforms current state-of-the-art novel view synthesis methods on most metrics of the Tanks and Temples dataset [16] and the MVImgNet [17] dataset, and our approach can render more photorealistic visual effects.

II. RELATED WORK

A. Sparse Input Novel-View Synthesis

NeRF-based [1] and 3DGS-based [2] methods require a large number of input images for scene reconstruction, which limits their practical applications. Currently, the academic community has proposed many solutions to this problem. RegNeRF [18] enhances the geometric accuracy of reconstructed scenes through a depth smoothness technique. FSGS [5] proposes a Gaussian densification strategy and pseudo view augmentation to improve the density and quality of Gaussian representations. DNGaussian [6] introduces a global-local depth regularization strategy to recover fine-grained scene details. DietNeRF [7], SIDGaussian [8] impose supervision on the embedding space to constraint the rendered unseen views. Some methods [19], [20] leverage the powerful prior knowledge of diffusion models to enhance reconstruction quality. Although these methods demonstrate promising results in sparse-view reconstruction, they all assume that the camera extrinsic parameters are known.

B. Reconstruction with unknown camera poses

Reconstructing 3D scenes without relying on SfM preprocessing for camera extrinsic estimation has emerged as an active research frontier. The pioneer work in this area was INeRF [21], which predict camera poses by adopting a pre-trained NeRF to match keypoints. BARF [22] presents a coarse-to-fine positional encoding strategy that enables the joint optimization of camera poses and NeRF. NeRFmm [23] simultaneously optimizes camera intrinsics, extrinsics, and the NeRF during training. SiNeRF [24] employs SIREN [25] layers and innovative sampling strategy to mitigate the suboptimal joint optimization in NeRFmm. Nope-NeRF [26] and CF-3DGS [27] incorporate pre-trained depth estimation models to obtain depth information, thereby constraining the optimization of NeRF and 3DGS. InstantSplat [9] achieves rapid scene reconstruction by leveraging pre-trained geometry priors.

III. METHODOLOGY

A. Preliminary

1) 3D Gaussian Splatting: 3D Gaussian Splatting (3DGS) [2] represents a 3D scene using a set of explicit Gaussian primitives. Each Gaussian primitive is defined by its center $\mu \in \mathbb{R}^3$ and covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$. The covariance matrix Σ is decomposed into a scaling factor $s \in \mathbb{R}^3$ and a rotation factor $r \in \mathbb{R}^4$. The k-th primitive G_k is defined as:

$$G_k(x) = e^{-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)} \tag{1}$$

where $\mu_k \in \mathbb{R}^3$ is the center of the Gaussian, and $\Sigma_k \in \mathbb{R}^{3\times3}$ is the covariance matrix. Each Gaussian primitive also retains a M-dimensional color feature $c \in \mathbb{R}^M$ and an opacity value $\alpha \in \mathbb{R}$. The color of pixel $\mathrm{x_p}$ on the image plane is calculated by the following formula:

$$\mathrm{C}(\mathrm{x_p}) = \sum_{k=1}^{N} c_k\alpha_k \prod_{j=1}^{k-1}(1-\alpha_j) \tag{2}$$

where N is the number of Gaussians contributing to pixel $\mathrm{x_p}$, c_k is the color of the k-th Gaussian, and α_k is the opacity of the Gaussian multiplied by the density of its projected 2D Gaussian distribution at the pixel location. Similarly, depth map and normal map are rendered by follows:

$$\mathrm{D}(\mathrm{x_p}) = \sum_{k=1}^{N} d_k\alpha_k \prod_{j=1}^{k-1}(1-\alpha_j) \tag{3}$$

$$\mathrm{N}(\mathrm{x_p}) = \sum_{k=1}^{N} n_k\alpha_k \prod_{j=1}^{k-1}(1-\alpha_j) \tag{4}$$

where d_i and n_i is the depth and normal direction of the i-th Gaussian. The optimization process refines the parameters of the Gaussians through gradient descent, guided by groundtrurh images. This explicit representation allows 3DGS to achieve high-quality novel view synthesis with reduced computational overhead compared to traditional methods.

2) MASt3R: MASt3R [10] is a Transformer-based framework for 3D reconstruction and image matching. Given two images I_1 and I_2, MASt3R processes them through a shared ViT [11] encoder to generate token representations F_1 and F_2. These tokens are jointly processed by Transformer decoders with cross-attention, enabling the network to understand spatial relationships and global 3D geometry. The outputs are fed into regression heads to produce dense pointmaps $\mathrm{P}_{1,1}$ and $\mathrm{P}_{2,1}$, which map each pixel to a 3D point in the scene, along with confidence maps $\mathrm{O}_{1,1}$ and $\mathrm{O}_{2,1}$ that indicate prediction reliability. The subscripts of $\mathrm{P}_{1,1}$ and $\mathrm{P}_{2,1}$ indicate that the pointmaps generated by I_1 and I_2 are both represented in the coordinate symtem of I_1. MASt3R employs a regression loss function to optimize the pointmap predictions:

$$\mathcal{L}_{reg} = \left\| \frac{1}{z_i} \cdot \mathrm{P}_{v,1} - \frac{1}{z_i} \cdot \hat{\mathrm{P}}_{v,1} \right\| \tag{5}$$

where $\mathrm{P}_{v,1}$ and $\hat{\mathrm{P}}_{v,1}$ are the predicted and ground-truth pointmaps, respectively, and z_i is a normalization factor. This loss function ensures the accuracy of the pointmap predictions. We use the pointmaps generated by MASt3R to initial 3DGS.

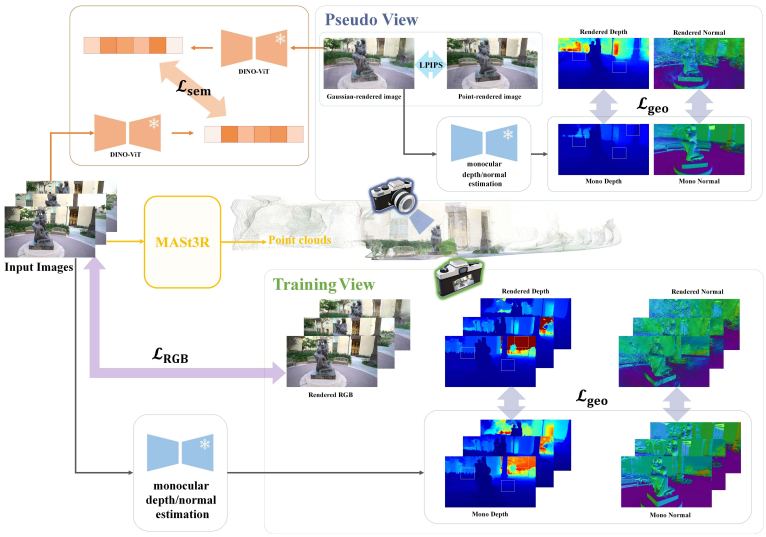

Fig. 1. **The framework of our method**. Our framework starts from uncalibrated sparse input images. By inputting images into MASt3R [10], we obtain initial point clouds and camera poses. We use these point clouds to initialize the 3DGS [2] model and adjust the camera poses during the training process. To enhance the rendering quality of 3DGS under sparse-view conditions, we designed several regularization methods to provide additional constraints, such as point-rendered LPIPS regularization, geometric regularization, and semantic regularization. We employ local depth regularization to enhance the details of the reconstructed scene.

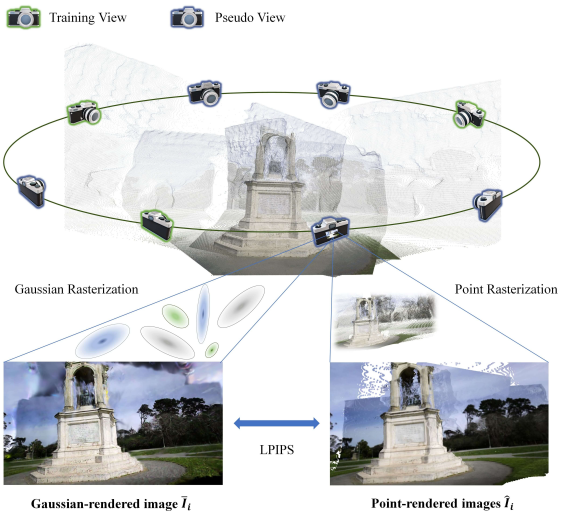

Fig. 2. We first interpolate the camera poses of the training images to generate pseudo camera poses. Then, under the same pseudo view i, we render both a Gaussian-rendered image $\bar{I}_i$ using Equation 2 and a point-rendered image $\hat{I}_i$ using Equation 6. The LPIPS of $\bar{I}_i$ and $\hat{I}_i$ is calculated as the point-rendered LPIPS regularization.

B. Proposed method

3DGS [2] fails to reconstruct from sparse-view due to the lack of sufficient supervisory signals, the algorithm is prone to overfitting to the sparse input images. To address this, we apply pseudo views to provide additional supervision. The poses of these pseudo views are obtained by interpolating the camera poses of the training images. Pseudo views will be utilized in point-rendered LPIPS regularization (III-B.1), geometric regularization (III-B.2), and semantic regularization (III-B.3). The overall framework of our method is shown in Figure 1.

1) Point-rendered LPIPS regularization: To address the overfitting issue of 3DGS under sparse-view inputs, we introduce point-rendered images as additional supervision to enhance the generalization capability of 3DGS on unseen viewpoints. Specifically, we apply point-rendered to the point cloud generated by MASt3R on pseudo viewpoints i, obtaining point-rendered images $\widehat{I}_i$. The color of each pixel $\mathrm{x_p}$ is determined by the K nearest 3D points to pixel $\mathrm{x_p}$, with weights assigned to these 3D points based on their distances from pixel $\mathrm{x_p}$ [14]. The point-rendered process can be formulated as follows:

$$\mathrm{C}(\mathrm{x_p}) = \sum_{k=1}^{K} c_k w_k, \quad w_k = \frac{e^{-d_k}}{\sum_{k=0}^{K-1} e^{-d_k}} \tag{6}$$

Here, $\mathrm{C}(\mathrm{x_p})$ represents the color of pixel $\mathrm{x_p}$ by using point-rendered, c_k is the color of the k-th 3D point, w_k is the corresponding weight, and d_k denotes the distance of the k-th 3D point from pixel $\mathrm{x_p}$. If no 3D points can be projected onto pixel $\mathrm{x_p}$, the color of pixel $\mathrm{x_p}$ is set to white.

We use Gaussian-rendered RGB image in the same pseudo view i to obtain $\bar{I}_i$, and then compute the LIPIS [12] between $\widehat{I}_i$ and $\bar{I}_i$.

$$\mathcal{L}_{\mathrm{pr}} = \mathrm{LPIPS}(\widehat{I}_i, \bar{I}_i) \tag{7}$$

This regularization term effectively avoids overfitting and renders more photorealistic images in unseen views. The process of point-rendered LPIPS regularization is shown in Figure 2.

2) Geometric regularization: **Local Depth Regularization.** The depth map's rich depth information offers robust guidance for converting target scenes from 2D images to 3D space. This enhances the 3D structural modeling of target scenes, making it more consistent with geometric and physical conditions. We acquire monocular depth maps via

a pre-trained monocular depth estimation model. Due to scale inconsistency between the depth maps predicted by the monocular model and the point clouds generated by MASt3R, directly applying a loss such as mean squared error(MSE) would not work well [20]. To address the scale ambiguity, we leverage the Pearson Correlation Coefficient [28] to measure the different between monocular depth and rendered depth. The Pearson Correlation is formulated as:

$$\mathrm{Corr}(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}} \tag{8}$$

where Cov denotes covariance, and Var represents the variance.

Prior works [5] primarily utilize global depth cues to enhance reconstruction quality. However, relying solely on global depth information biases the model toward capturing global features while neglecting fine-grained local geometries. Similar to LMGaussian [14], we propose to decompose depth maps into local patches and focus on inter-patch correlations to improve detail preservation. Specifically, we divide both the monocular depth map D and rendered depth map $\overline{D}$ into L non-overlapping patches and randomly select M patches during each iteration. The proposed depth correlation loss is defined as:

$$\mathcal{L}_{\mathrm{depth}} = \frac{1}{M}\sum_{m=1}^{M} 1 - \mathrm{Corr}(P_m, \overline{P}_m) \tag{9}$$

where P_m and $\overline{P}_m$ denote the m-th patch from D and $\overline{D}$, respectively. This loss is applied to both training views and pseudo views to ensure consistent depth alignment across different viewpoints.

Normal Regularization. Inconsistent surface normals across different viewpoints can degrade reconstruction quality. Monocular normal estimation models provide strong geometric priors to effectively resolve multi-view normal inconsistencies. Since normal maps are free from scale ambiguity, we directly constrain the rendered normals using angular loss and L1 loss:

$$\mathcal{L}_{\mathrm{normal}} = \sum_{i=0}^{HW} \left\|N(i) - \overline{N}(i)\right\|_1 + \left\|1 - N(i)^\top \overline{N}(i)\right\|_1 \tag{10}$$

where $N(i)$ and $\overline{N}(i)$ represent i-th pixel value of monocular normal and rendered normal map, respectively. Normal regularization is similarly apply on both training views and pseudo views. We define the geometric regularization as the sum of the depth loss and the normal loss:

$$\mathcal{L}_{\mathrm{geo}} = \beta_{depth}\mathcal{L}_{\mathrm{depth}} + \beta_{normal}\mathcal{L}_{normal} \tag{11}$$

where β_{depth} and β_{normal} denote the weight of $\mathcal{L}_{\mathrm{depth}}$ and $\mathcal{L}_{normal}$.

3) Semantic regularization: Prior research [7], [8], [29], [30] has demonstrated that pretrained multimodal models such as CLIP [31] and DINO-ViT [15] can effectively improve the quality of reconstruction. These models encode input images into semantic-rich feature vectors. In this work, we adopt DINO-ViT for semantic regularization. Specifically, both training-view image and rendered images from pseudo viewpoints are fed into the DINO-ViT model to generate their corresponding semantic embeddings. The semantic loss is then computed as the L2 distance between these two embeddings, formulated as:

$$\mathcal{L}_{\mathrm{sem}} = \left\|\mathrm{encode}(\overline{I}_i) - \mathrm{encode}(I_k)\right\|^2 \tag{12}$$

where $\mathrm{encode}(\overline{I}_i)$ and $\mathrm{encode}(I_k)$ denote the semantic embeddings of the i-th pseudo-view rendered image and k-th training-view input image, respectively. Semantic regularization encourages the model to render semantically consistent views across different viewpoints by aligning the latent representations of synthesized and ground-truth images.

4) Total loss: Our proposed total loss function is formulated as:

$$\mathcal{L}_{\mathrm{total}} = \beta_{\mathrm{pr}}\mathcal{L}_{\mathrm{pr}} + \mathcal{L}_{\mathrm{geo}} + \beta_{\mathrm{sem}}\mathcal{L}_{\mathrm{sem}} + \mathcal{L}_{\mathrm{rgb}} \tag{13}$$

where $\mathcal{L}_{\mathrm{rgb}} = (1-\lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{\text{D-SSIM}}$ is the same loss proposed by vanilla 3DGS [2].

C. Other details

Point clouds downsampling. The point clouds generated by MASt3R [10] often suffer from floating object, artifacts, and distortion. Furthermore, since MASt3R produces pixel-aligned point clouds, shared foreground regions across multiple views tend to generate redundant point clouds, leading to inefficient scene representation. To enhance robustness, we adopt a downsampling strategy inspired by InstantSplat [9]. The process begins by ranking input views based on confidence scores derived from MASt3R's per-pixel confidence maps, where the confidence score s_i for view i is computed by the average of its confidence values.

$$s_i = \frac{1}{|\mathrm{O}_i|}\sum_{o\in \mathrm{O}_i} o \tag{14}$$

where O_i represents confidence map of view i and o represents pixel value of confidence map. Views with higher confidence scores are prioritized as they encapsulate more reliable geometric information. Starting from the lowest-confidence view j, point clouds from higher-confidence views are projected onto view j's coordinate system to generate a fused point set.

$$\mathrm{D}_{\mathrm{proj},j} = \bigcap_{i|s_i>s_j} \mathrm{Proj}_j(\widetilde{\mathrm{P}}_i) \tag{15}$$

where $\widetilde{\mathrm{P}}_j$ represents the point cloud generated by MASt3R corresponding to input image i, Proj represents the function applying the projection operator across all views i where $s_i >= s_j$. A depth verification step is then applied to retain only points where the discrepancy between the projected depth and the original depth from MASt3R falls below a predefined threshold θ.

$$\mathcal{M}_j = \begin{cases} 1 & \text{if } |\mathrm{D}_{\mathrm{proj},j} - \mathrm{D}_{\mathrm{orig},j}| < \theta \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

This iterative procedure, executed across all views in ascending order of confidence, produces binary masks $\mathcal{M}_j$ to filter redundant points, resulting in a refined point cloud P_j.

$$\mathrm{P}_j = (1 - \mathcal{M}_j) \cdot \widetilde{\mathrm{P}}_j \tag{17}$$

Camera Pose Refinement. The camera extrinsics estimated by MASt3R often contain inaccuracies, leading to geometric inconsistencies when directly used for model training. To address this, we jointly optimize the camera extrinsics alongside the 3D Gaussian parameters during training, inspired by the joint optimization strategy in InstantSplat [9]. In traditional benchmark datasets, camera poses and calibration parameters are typically precomputed through global optimization (e.g., COLMAP) with access to all training and testing views. However, in a pose-free sparse-view setting, the camera poses for test views must be regressed via self-supervised optimization. Inspired by INeRF[21] and InstantSplat [9], our approach fixes the 3D Gaussians parameters G^* learned during training and optimizes only the extrinsic parameters T_{test} of the test views. The optimization objective is formulated as:

$$T^*_{\text{test}} = \underset{T_{\text{test}}}{\operatorname{argmin}} \sum_{i=1}^{HW} \left\| \mathrm{C}^i(G^*, T_{\text{test}}) - \mathrm{C}^i_{\text{test}} \right\| \tag{18}$$

where $\mathrm{C}(\cdot)$ denotes the differentiable rendering function based on Equation 2, which projects the scene G^* into the image plane under camera pose T_{test} and C_{test} represents the ground-truth images of test view. The optimized poses T^*_{test} are then used to evaluate visual quality metrics and pose estimation accuracy.

IV. Experiments

A. Experimental Setup

Datasets. We evaluated our method on Tanks and Temples [16] dataset and MVImgNet [17] dataset. Tanks and Temples features complex large-scale scenes for 3D reconstruction evaluation, while MVImgNet provides diverse multi-view imagery of common daily objects to support multi-task vision applications. We evaluated our method on eight scenes from the Tanks and Temples dataset and seven scenes from the MVImgNet dataset by following previous research [9]. We select 3, 6 and 12 images from these two datasets to satisfy the sparse-view settings.

Train/Test Datasets Split. Following InstantSplat's [9] strategy, we employed a consistent uniform sampling approach across both the Tanks and Temples and MVImgNet datasets. From each dataset, 12 images were selected for testing by uniformly sampling frames while excluding the first and last frames. An equivalent number of training images were then chosen from the remaining frames. This sampling methodology ensures comprehensive coverage of the entire image set while maintaining temporal separation between training and test frames.

Metrics. Our method is evaluated on standard benchmarks through two primary tasks: novel view synthesis and camera pose estimation. For novel view synthesis, performance is quantified using established metrics: Structural Similarity Index Measure (SSIM) [32], Learned Perceptual Image Patch Similarity (LPIPS) [12] and Peak Signal-to-Noise Ratio (PSNR). For camera pose estimation, we adopt the Absolute Trajectory Error (ATE) metric as defined in [26], using COLMAP poses from all dense views as ground-truth references.

Baselines. Our comparisons on pose-free methods include Nope-NeRF [26], CF-3DGS [27], NeRFmm [23] and InstantSplat [9]. Additionally, we compared our method with 3DGS [2] and FSGS [5], both of which use COLMAP to precompute camera parameters.

B. Implementation Details

Our work is based on InstantSplat [9]. We implemented our entire framework in PyTorch. Our experiment was conducted on a single Nvidia A100 GPU. We use Metric3D [13] to predict monocular depth maps and monocular normal maps for all input views. We train the model with 1000 iterations for all datasets. Specifically, point-rendered LPIPS [12] regularization and semantic regularization were applied from the 100th to the 800th iterations, while geometric regularization was used from the 200th to the 700th iterations. We set the parameters β_{pr}=0.25, β_{depth}=0.05, β_{normal}=0.01, β_{sem}=0.5 in the loss functions for all experiments. These parameters were determined through experimental validation. During evaluation, we used 500 iterations for test view optimization.

C. Experimental Results and Analysis

We evaluate novel view synthesis and pose estimation on the Tanks and Temples [16] and MVImgNet [17] datasets, with results summarized in Tables I, II, and Figure 3. The data in Tables I and II are sourced from the reported results of InstantSplat [9].

Nope-NeRF [26] demonstrates promising pose estimation accuracy in reconstruction tasks with dense video sequences. However, its pose estimation precision degrades under sparse-view settings, and due to insufficient geometric constraints, Nope-NeRF tends to render blurry images. Additionally, Nope-NeRF suffers from prolonged inference times, failing to achieve real-time rendering. Similarly, CF-3DGS [27] requires dense video sequences as input. Under sparse-view settting, CF-3DGS generates novel views with artifacts caused by inaccurate pose estimation, as shown in figure 3. NeRFmm [23] aims to jointly optimize camera parameters and NeRF [1], yet it often struggles to achieve satisfactory results due to inherent challenges in naive joint optimization. InstantSplat [9] focuses on rapid scene reconstruction but suffers from geometric inaccuracies due to insufficient supervision. Furthermore, InstantSplat produces non-photorealistic visual effect in scenarios with large viewpoint variations.

In Table III, we provide a detailed comparison between our method and InstantSplat on novel view synthesis metrics (SSIM, PSNR, LPIPS) across eight scenes from Tanks and Temples and seven scenes from MVImgNet. Benefiting from

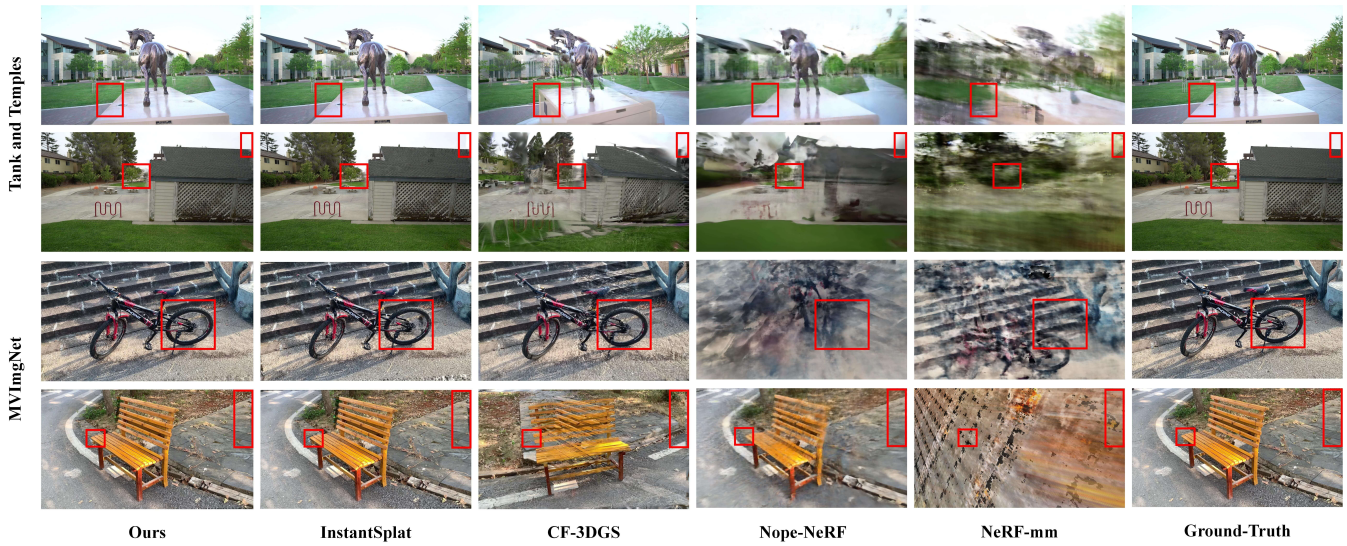

Fig. 3. **Qualitative comparison** between our method and various pose-free baseline method on Tanks and Temples dataset [16] and MVImgNet dataset [17] under 3-view setting. Compared to other baseline methods, our method can render more photorealistic visual effects.

TABLE I
QUANTITATIVE EVALUATIONS ON TANKS AND TEMPLES DATASET.

	SSIM			LPIPS			ATE		
	3-view	6-view	12-view	3-view	6-view	12-view	3-view	6-view	12-view
COLMAP+3DGS [2]	0.3755	0.5917	0.7163	0.5130	0.3433	0.2505	-	-	-
COLMAP+FSGS [5]	0.5701	0.7752	0.8479	0.3465	0.1927	0.1477	-	-	-
NoPe-NeRF [26]	0.4570	0.5067	0.6096	0.6168	0.5780	0.5067	0.2828	0.1431	0.1029
CF-3DGS [27]	0.4066	0.4690	0.5077	0.4520	0.4219	0.4189	0.1937	0.1572	0.1031
NeRF-mm [23]	0.4019	0.4308	0.4677	0.6421	0.6252	0.6020	0.2721	0.2329	0.1529
InstantSplat [9]	0.7615	0.8453	0.8785	0.1634	0.1173	0.1068	0.0189	0.0164	**0.0101**
Ours	**0.7748**	**0.8486**	**0.8850**	**0.1583**	**0.1166**	**0.1029**	**0.0186**	**0.0162**	**0.0101**

TABLE II
QUANTITATIVE EVALUATIONS ON MVIMGNET DATASET.

	SSIM			LPIPS			ATE		
	3-view	6-view	12-view	3-view	6-view	12-view	3-view	6-view	12-view
NoPe-NeRF [26]	0.4326	0.4329	0.4686	0.6674	0.6614	0.6257	0.2780	0.1740	0.1493
CF-3DGS [27]	0.3414	0.3544	0.3655	0.5523	0.4326	0.4492	0.1593	0.1981	0.1243
NeRF-mm [23]	0.3752	0.3685	0.3718	0.7001	0.6252	0.6020	0.2721	0.2376	0.1529
InstantSplat [9]	0.5628	0.6933	0.7321	0.3688	0.2611	**0.2421**	0.0184	**0.0259**	0.0164
Ours	**0.5736**	**0.6946**	**0.7385**	**0.3450**	**0.2533**	0.2431	**0.0182**	**0.0259**	**0.0162**

our proposed regularization techniques, our method outperforms InstantSplat in most scenes. Figure 4 further visualizes the rendered RGB images, depth maps, and normal maps of both methods. Experimental results demonstrate that our method achieves significant improvements in rendered depth and normal maps compared to InstantSplat, this means that our method can reconstruct more accurate geometric quality.

Notably, while our method prioritizes novel view synthesis quality, it achieves pose estimation accuracy that matches or even surpasses InstantSplat in most scenes.

D. Ablation Study

Here, we evaluate the effect of various components of our method on the reconstruction quality, both qualitative (Fig. 5) and quantitative (Table. IV). The quantitative results are obtained on the Tanks and Temples dataset with 3 input views. Our proposed point-rendered LPIPS regularization enables rendering visually plausible results when there are large differences in viewpoints, while the geometric regularization and semantic regularization mitigate artifacts around object boundaries and alleviate visual quality degradation caused by floaters. Figure 5 and Table IV indicate that each components we proposed is effective for improving the effect of reconstruction.

V. CONCLUSIONS

In this paper, we present a novel approach for high-quality sparse-view reconstruction without requiring camera extrinsic parameters. To address the challenge of unknown camera extrinsics, we first employ the MASt3R [10] model to estimate initial camera parameters. Secondly, we employ three key regularization methods to enhance the quality

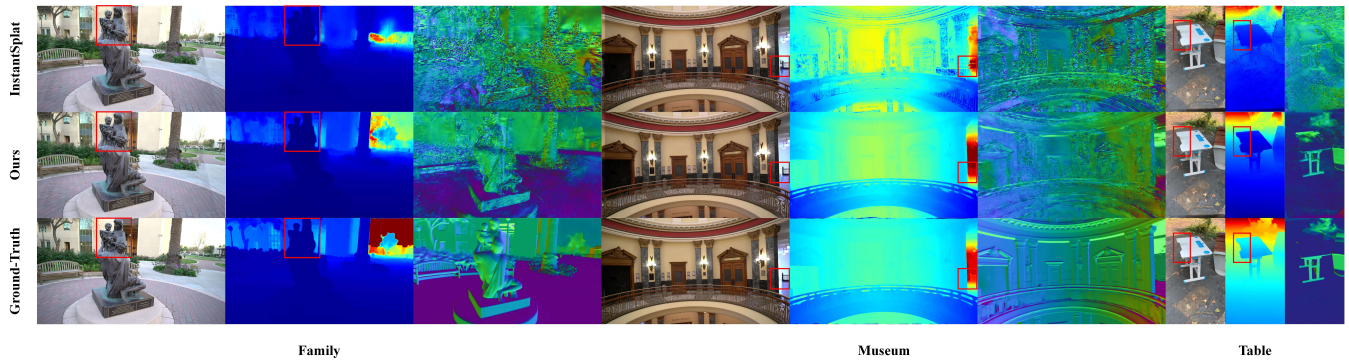

Fig. 4. We compare the RGB (left), depth maps(middle), and normal maps (right) rendered by our method and InstantSplat across three scenes. The Family and Museum scenes are from the Tanks and Temples dataset [16], while the Table scene is from the MVImgNet dataset [17]. We use depth maps and normal maps generated by Metric3D [13] as ground-truth. The depth maps and normal maps rendered by our method significantly outperform those of InstantSplat. Taking the Family scene as an example, as indicated by the red box in the first column, our method successfully eliminates artifacts around objects with the guidance of geometric regularization.

TABLE III

WE ADDITIONALLY PROVIDE A DETAILED COMPARISON WITH INSTANTSPLAT [9] ON 8 SCENES FROM THE TANKS AND TEMPLES DATASET AND 7 SCENES FROM THE MVIMGNET DATASET. THE INSTANTSPLAT RESULTS IN THIS TABLE WERE OBTAINED BY REPRODUCING THEIR METHOD; EXPERIMENTS WERE PERFORMED UNDER THE 3-VIEW SETTING.

Tanks and Temples Datasets

	Scene	Ballroom	Barn	Church	Family	Francis	Horse	Ignatius	Museum
SSIM	InstantSplat [9]	0.8408	0.6828	0.6711	0.8351	0.7665	0.8210	0.7020	0.7867
	Ours	**0.8479**	**0.7064**	**0.6817**	**0.8437**	**0.7810**	**0.8293**	**0.7125**	**0.7963**
PSNR	InstantSplat	24.5704	21.0156	19.1130	23.5606	23.6986	23.2684	22.7409	22.4587
	Ours	**24.8816**	**21.9713**	**19.3834**	**24.7808**	**24.7053**	**23.9233**	**23.1407**	**22.9840**
LPIPS	InstantSplat	0.1038	0.2099	0.2250	0.1148	0.2374	0.1273	0.1743	**0.1280**
	Ours	**0.1006**	**0.1968**	**0.2239**	**0.1103**	**0.2227**	**0.1197**	**0.1638**	0.1288

MVImgNet Datasets

	Scene	Bench	Bicycle	Car	Chair	Ladder	SUV	Table
SSIM	InstantSplat	0.4034	0.3357	0.8288	0.5734	0.3345	0.8149	0.5469
	Ours	**0.4321**	**0.3624**	**0.8399**	**0.5974**	**0.3512**	**0.8275**	**0.6047**
PSNR	InstantSplat	16.5087	14.9035	22.3987	17.5935	15.1154	22.4080	15.4630
	Ours	**17.6699**	**15.7975**	**23.6603**	**18.4807**	**15.8348**	**24.0443**	**17.4602**
LPIPS	InstantSplat	0.3934	0.4347	0.1795	0.4768	0.4690	0.2047	0.4456
	Ours	**0.3620**	**0.4214**	**0.1707**	**0.4146**	**0.4590**	**0.2012**	**0.3858**

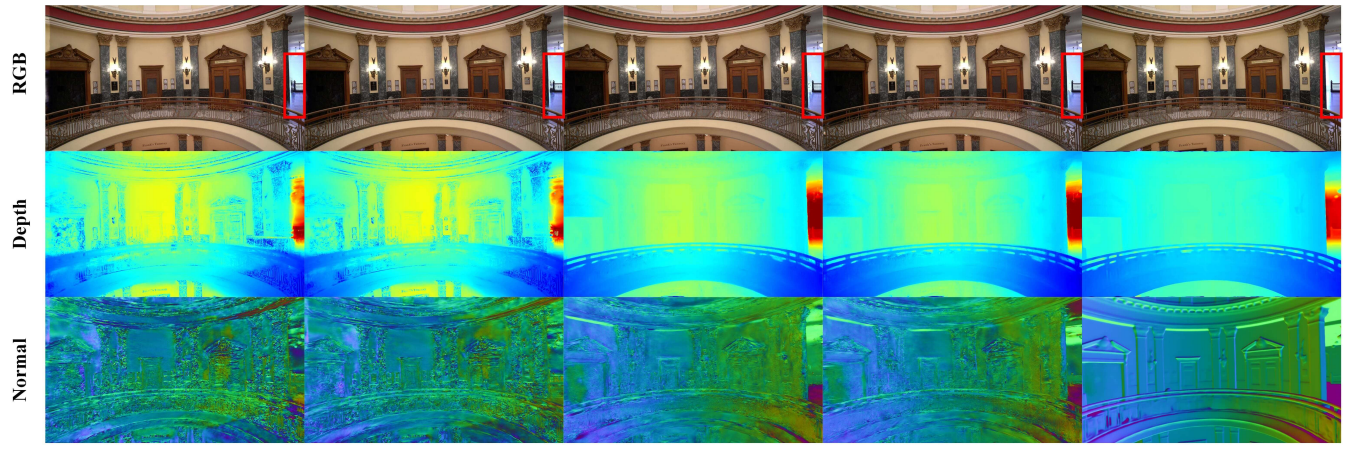

Fig. 5. We visualize the ablation study on Museum from Tanks and Temples dataset [16]; the study was performed under 3-view setting. Qualitative results demonstrate that each proposed component contributes to enhanced visual quality in novel view synthesis.

TABLE IV

WE PROVIDE QUANTITATIVE RESULTS OF OUR METHOD WITH (✓) OR WITHOUT (✗) PROPOSED COMPONENTS.

$\mathcal{L}_{pr}$	$\mathcal{L}_{geo}$	$\mathcal{L}_{sem}$	SSIM ↑	PSNR ↑	LPIPS ↓
✗	✗	✗	0.7335	22.0186	0.1965
✓	✗	✗	0.7429	22.3485	0.1724
✓	✓	✗	0.7712	22.8921	0.1601
✓	✓	✓	**0.7748**	**23.2213**	**0.1583**

of sparse-view reconstruction: 1) We apply Point-rendered LPIPS regularization to prevent overfitting in sparse-view; 2) We use geometric regularization to mitigate geometric degradation and strengthen the detail of reconstructed scenes; 3) We leverage semantic regularization to ensure semantic consistency of the scene across different viewpoints. Experimental results demonstrate that our method surpasses state-of-the-art approaches in terms of novel view synthesis metrics (SSIM [32], LPIPS [12], PSNR) and camera pose estimation metrics (ATE) on both the Tanks and Temples dataset and the MVImgNet dataset.

ACKNOWLEDGMENT

This research was supported by the Baima Lake Laboratory Joint Funds of the Zhejiang Provincial Natural Science Foundation of China under Grant No.LBMHD25F030001, in part by NSFC 62088101 Autonomous Intelligent Unmanned Systems.

REFERENCES

[1] 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] 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.

[4] H. Li, J. Li, D. Zhang, C. Wu, J. Shi, C. Zhao, H. Feng, E. Ding, J. Wang, and J. Han, "Vdg: Vision-only dynamic gaussian for driving simulation," *IEEE Robotics and Automation Letters*, vol. 10, no. 5, pp. 5138–5145, 2025.

[5] Z. Zhu, Z. Fan, Y. Jiang, and Z. Wang, "Fsgs: Real-time few-shot view synthesis using gaussian splatting," in *European Conference on Computer Vision*, 2025.

[6] J. Li, J. Zhang, X. Bai, J. Zheng, X. Ning, J. Zhou, and L. Gu, "Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization," *IEEE*, 2024.

[7] A. Jain, M. Tancik, and P. Abbeel, "Putting nerf on a diet: Semantically consistent few-shot view synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5885–5894.

[8] Z. He, Z. Xiao, K.-C. Chan, Y. Zuo, J. Xiao, and K.-M. Lam, "See in detail: Enhancing sparse-view 3d gaussian splatting with local depth and semantic regularization," *arXiv preprint arXiv:2501.11508*, 2025.

[9] Z. Fan, K. Wen, W. Cong, K. Wang, J. Zhang, X. Ding, D. Xu, B. Ivanovic, M. Pavone, G. Pavlakos, *et al.*, "Instantsplat: Sparse-view sfm-free gaussian splatting in seconds," *arXiv preprint arXiv:2403.20309*, 2024.

[10] 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.

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[12] 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.

[13] W. Yin, C. Zhang, H. Chen, Z. Cai, G. Yu, K. Wang, X. Chen, and C. Shen, "Metric3d: Towards zero-shot metric 3d prediction from a single image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 9043–9053.

[14] H. Yu, X. Long, and P. Tan, "Lm-gaussian: Boost sparse-view 3d gaussian splatting with large model priors," *arXiv preprint arXiv:2409.03456*, 2024.

[15] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," *arXiv preprint arXiv:2203.03605*, 2022.

[16] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.

[17] X. Yu, M. Xu, Y. Zhang, H. Liu, C. Ye, Y. Wu, Z. Yan, C. Zhu, Z. Xiong, T. Liang, *et al.*, "Mvimgnet: A large-scale dataset of multi-view images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 9150–9161.

[18] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. M. Sajjadi, A. Geiger, and N. Radwan, "Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs," *arXiv e-prints*, 2021.

[19] Q. Wang, Y. Zhao, J. Ma, and J. Li, "How to use diffusion priors under sparse views?" *arXiv preprint arXiv:2412.02225*, 2024.

[20] H. Xiong, *SparseGS: Real-time 360° sparse view synthesis using Gaussian splatting*. University of California, Los Angeles, 2024.

[21] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "inerf: Inverting neural radiance fields for pose estimation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1323–1330.

[22] 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.

[23] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu, "Nerf–: Neural radiance fields without known camera parameters," 2021.

[24] Y. Xia, H. Tang, R. Timofte, and L. Van Gool, "Sinerf: Sinusoidal neural radiance fields for joint pose estimation and scene reconstruction," *arXiv preprint arXiv:2210.04553*, 2022.

[25] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in neural information processing systems*, vol. 33, pp. 7462–7473, 2020.

[26] 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.

[27] Y. Fu, S. Liu, A. Kulkarni, J. Kautz, A. A. Efros, and X. Wang, "Colmap-free 3d gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 796–20 805.

[28] I. Cohen, Y. Huang, J. Chen, J. Benesty, J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," *Noise reduction in speech processing*, pp. 1–4, 2009.

[29] L. Szilagyi, F. Engelmann, and J. Bohg, "Slag: Scalable language-augmented gaussian splatting," *IEEE Robotics and Automation Letters*, vol. 10, no. 7, pp. 6991–6998, 2025.

[30] J. Zeng, Q. Ye, T. Liu, Y. Xu, J. Li, J. Xu, L. Li, and J. Chen, "Multi-robot autonomous 3d reconstruction using gaussian splatting with semantic guidance," *IEEE Robotics and Automation Letters*, vol. 10, no. 6, pp. 5617–5624, 2025.

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

[32] 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.