

UniqueSplat: View-conditioned 3D Gaussian Splatting for Generalizable 3D Reconstruction

Haixu Song, Xiaoke Yang, Shengjun Zhang, Jiwen Lu, *Fellow, IEEE*, Yueqi Duan, *Member, IEEE*

Abstract—In this paper, we propose UniqueSplat, a view-conditioned feed-forward 3D Gaussian Splatting model to reconstruct customized 3D radiance fields for each view query. Existing feed-forward methods such as pixelSplat and MVSplat aim to generate fixed Gaussians across all views of each scene by minimizing the error between rendered views and ground-truth images. However, such fixed Gaussians generally render images from all views and lack the ability to adapt to specific viewpoints, as they do not incorporate target view information when predicting Gaussians. To address this, our UniqueSplat learns the view-conditioned information as a prior and incorporates this knowledge into network parameters, so that Gaussians are dynamically adjusted in accordance with different views. Specifically, we propose a two-branch view-conditioned hyperNetwork to simultaneously learn view-agnostic embeddings and view-specific knowledge, which not only explores the shareable knowledge from various views, but also adapts the model to specific views at test time. Extensive experiments on widely-used datasets including RealEstate10K, ACID and DTU demonstrate the superiority of UniqueSplat over the state-of-the-art methods. Moreover, UniqueSplat encouragingly outperforms existing methods in cross-dataset evaluation, showing its notable generalization ability.

Index Terms—Feed-forward 3D Gaussian Splatting, view-conditioned, sparse-view, generalizable novel view synthesis.

I. INTRODUCTION

NOVEL view synthesis is a crucial problem in computer vision due to its broad applications, such as virtual reality [2], robotics [3] and autonomous driving [4]. Traditional approaches represent scenes using point clouds [5], meshes [6], [7], or a 4D light field function [8]. Neural implicit representations have achieved great success in recent years, while such methods typically require substantial computational resources for both training and rendering, which limits their efficiency and scalability [9]–[13]. Recently, 3D Gaussian Splatting (3DGS) [14] has drawn increasing attention which represents the scene with a set of 3D Gaussian primitives to enable real-time rendering. This approach leverages rasterization-based rendering to avoid dense point querying, thereby maintaining both high efficiency and quality.

Nevertheless, 3DGS requires substantial images to optimize each specific scene, which presents challenges for generalizable 3D reconstruction across diverse scenes [15], [16]. To address this, several generalizable methods have been proposed to improve generalizability and transferability from

sparse view images [1], [17]–[19]. These methods utilize feed-forward networks to predict the parameters of pixel-aligned Gaussians in single forward pass, enhancing both efficiency and applicability in reconstructing unseen scenes. For instance, pixelSplat [20] employs epipolar transformers to unproject depth maps from corresponding views to predict the parameters of 3D Gaussians, while MVSplat [1] leverages cost volume representation to avoid the noisy artifacts.

While these methods have achieved encouraging performance, they focus on learning a feed-forward model by minimizing the loss between rendered views and ground-truth images, aiming to predict fixed Gaussians with the minimal average error across all views for each scene. However, while such fixed Gaussians generally render images from multiple views, they cannot specialize to a particular viewpoint, since the target view information is not incorporated into the Gaussian prediction process. As illustrated in Figure 1, MVSplat exhibits issues such as cracks, blurs and artifacts when synthesizing novel views. In this paper, we propose UniqueSplat, a view-conditioned feed-forward 3D Gaussian Splatting model that reconstructs customized 3D radiance fields for each view query. Unlike existing methods that predict fixed Gaussians for rendering all views of a scene, our UniqueSplat learns view-conditioned information as a prior and integrates the knowledge into the network parameters to customize 3D radiance fields, enabling the Gaussian fields to dynamically adjust for each specific view. Specifically, instead of directly relying on the input views, we decompose the view-conditioned knowledge into two components for each view query: view-agnostic information, which remains consistent across views, and view-specific information, which varies for each view query in the scene. Therefore, we design a view-conditioned hypernetwork consisting of two branches: a view-agnostic branch and a view-specific branch, in contrast to prior hypernetwork-based NVS methods that adopt only a single branch. The view-agnostic branch maintains learnable embeddings which are continuously updated during training to abstract the characteristics across various views of the training scenes. Since the view-agnostic branch generates a fixed set of parameters to predict Gaussians across multiple views during testing, we introduce the view-specific branch as a complementary to dynamically adjust the predictor for each given view. To achieve this, we construct the feature representation via combining the features of input sparse-view images for each view query, and then project it into an embedding to incorporate view-specific knowledge. Finally, we fuse the weights generated by both branches to customize

Haixu Song, Xiaoke Yang, Shengjun Zhang, Jiwen Lu and Yueqi Duan are with Tsinghua University, China (e-mail: shx22@mails.tsinghua.edu.cn; yangxk22@mails.tsinghua.edu.cn; zhangsj23@mails.tsinghua.edu.cn; lujiwen@tsinghua.edu.cn; duanyueqi@tsinghua.edu.cn).

Corresponding author: Yueqi Duan

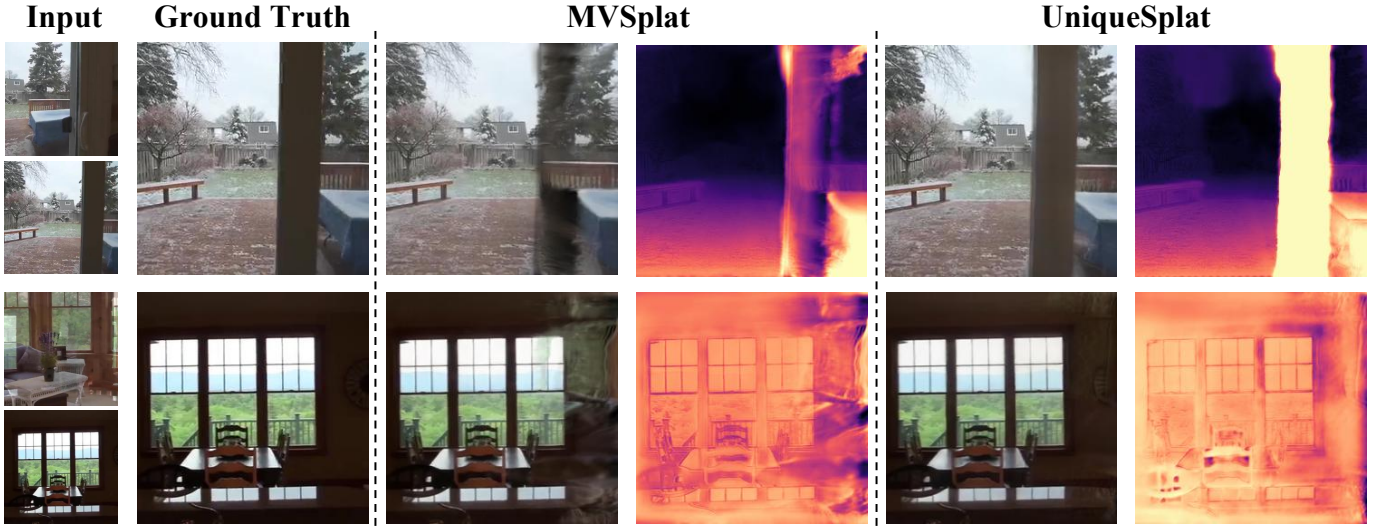


Fig. 1. Qualitative comparison between MVSplat [1] and UniqueSplat. UniqueSplat incorporates view-conditioned information as a prior to achieve superior quality in novel view synthesis and produce high-quality depth maps, while MVSplat presents issues such as cracks, blurs, and artifacts.

Gaussians for each individual view.

We summarize our key contributions as follows:

- We propose UniqueSplat, a method for novel view synthesis from sparse view images. While existing methods leverage fixed Gaussians to render all views, we are the first to customize Gaussians which dynamically adjust according to the view query.
- We design a dual-branch view-conditioned hypernetwork consisting of a view-agnostic branch and a view-specific branch to capture shareable knowledge and view-specific information, unlike prior NVS methods that employ hypernetworks to directly generate weights solely from input signals.
- We conduct extensive experiments on widely-used benchmark datasets, including RealEstate10K, ACID and DTU. Our method achieves state-of-the-art results in both intra-test and cross-test settings.

II. RELATED WORKS

A. Single-scene Novel View Synthesis

Neural rendering [10], [21] has attracted significant interest in recent years due to its impressive capability to synthesize high-quality novel views from 3D scenes. However, despite their remarkable visual fidelity, these methods often suffer from high computational costs. This issue arises because rendering a single image typically requires querying a neural field multiple times per ray, resulting in slow performance, particularly for real-time applications. To mitigate these costs, subsequent studies have introduced discrete data structures to accelerate the rendering process [9], [11], [12], [22], [23]. While these methods successfully accelerate the rendering process to some extent, they still face limitations in achieving real-time performance, particularly when rendering at high resolutions. More recently, 3D Gaussian Splatting (3DGS) [14] has emerged as a promising technique in novel view synthesis. By employing an efficient rasterization-based splatting

approach, 3DGS leverages a set of 3D Gaussian primitives to represent a scene. This novel technique offers considerable advancements in both rendering speed and quality, making it a compelling alternative to earlier methods that rely on dense neural representations. Despite these significant advantages, 3DGS can still produce artifacts during the splatting process, motivating ongoing research efforts to improve the quality and realism of rendered views [15], [16], [24], [25]. Another key challenge associated with 3DGS is its high storage requirement. To represent a single scene accurately, 3DGS often necessitates millions of parameters, leading to substantial memory and storage demands. To tackle this issue, several research efforts [26]–[28] focus on reducing memory usage to enable real-time rendering without compromising on quality of the rendered scenes. Additionally, some works [29], [30] aim to minimize the number of input images required for scene reconstruction, further enhancing the efficiency of 3DGS. Moreover, several studies [31]–[33] have explored extending the application of 3D Gaussians into the realm of 4D scenarios by incorporating dynamic elements that align with real-world physical behavior. Besides, recent works have also incorporated physical priors or novel neural representations to further improve rendering quality. 3DGS-Enhancer [34] leverages 2D video diffusion priors to address the challenging 3D view-consistency problem by reformulating it as achieving temporal consistency in video generation. Mvpgs [35] exploits multi-view priors to enhance the quality of geometric initialization for 3DGS and further introduces multiple constraints to improve rendering quality. LITA-GS [36] incorporates an illumination-invariant physical prior and develops a lighting-agnostic rendering strategy to achieve high-quality, normally exposed representations under adverse illumination conditions. In addition, Chung et al. [37] propose a depth-guided Gaussian Splatting optimization strategy that mitigates overfitting by refining scene optimization with depth priors. However, these methods primarily focus on optimizing a specific scene. When applied to a new scene, they require additional time for

optimization and lack generalization ability.

B. Generalizable Novel View Synthesis

Unlike optimization-based methods, which typically require dozens of minutes to train a separate model for each scene, feed-forward networks directly synthesize novel views in a single inference, requiring substantially less time compared with optimization-based methods. This results in superior cross-dataset generalizability and faster rendering speeds. In recent years, implicit representations have parameterized fully connected neural networks to represent complex 3D scenes with global and local codes [38]–[40]. Some approaches enhance the quality of neural radiance field reconstruction via constructing cost volumes at pre-defined reference view-points [41]–[43], while others perform independent ray-based rendering [44]–[47]. Additionally, there are methods specifically designed for large baseline settings [48], [49], which focus on reconstructing 3D scenes from a wider range of camera positions. Despite significant advancements in MLP-based representations, these methods are often slow to render, limiting their potential for real-time applications. To address this, a variety of faster representations have been developed, including voxel grid representations [50], [51] and triplane representations [52], [53]. Voxel grids significantly accelerate rendering by directly encoding opacities and colors, although they scale poorly with resolution. The triplane representation, on the other hand, enables view-space reconstruction and provides performance suitable for single-view reconstruction. In recent years, there has been a surge of interest in generalizable 3D Gaussian Splatting (3DGS) methods for reconstructing 3D scenes from sparse views. For instance, Splatter Image [17] and GPS-Gaussian [2] focus on regressing pixel-aligned Gaussian parameters to reconstruct specific objects or humans rather than complex scenes. In contrast, pixel-Splat [20] leverages sparse view inputs to predict Gaussian parameters by utilizing epipolar geometry and depth estimation. MVSPat [1] introduces a cost volume representation to directly predict depth from cross-view features, significantly improving geometric quality. LatentSplat [54] addresses reconstruction challenges in object-centric scenes with 360° views by incorporating variational Gaussians, improving both accuracy and completeness. TranSplat [18] applies depth-aware deformable matching and monocular depth priors to enhance reconstruction quality. However, these feed-forward methods generate fixed Gaussians across all views of a scene, failing to adapt to each individual view. To address this limitation, we propose UniqueSplat, which incorporates view-conditioned information to customize Gaussians for each view query, resulting in more realistic and high-quality images.

C. HypernetWorks

Hypernetworks [55] are a class of neural networks designed to generate weights for another neural network, known as the primary network, based on a specific input embedding. This input embedding can be derived from various types of data, allowing the hypernetwork to conditionally adjust the weights of the primary network depending on the context. Hypernetworks

have emerged as a powerful deep learning technique, enabling greater adaptability, dynamism and information sharing [56]. They have been integrated into models for tasks such as image recognition, semantic segmentation, and neural architecture search, where they replace convolutional or linear layers [57], [58]. In the context of implicit neural representations, hypernetworks have proven particularly useful for generating task-conditioned networks that can adapt to new data instances, making implicit representations more flexible for tasks where the scene or data is subject to change. Recently, hypernetworks have been successfully applied to sound [59], representation compression [60] and novel view synthesis [61]–[63]. However, such hypernetwork-based NVS methods primarily focus on generating implicit neural representation directly from input signals. In contrast, our proposed UniqueSplat employs a view-conditioned hypernetwork to explicitly model Gaussian representation, predicting dynamic Gaussians conditioned on the query view. By adopting a dual-branch design comprised of a view-specific branch and a view-agnostic branch, our model effectively disentangles scene knowledge, thereby achieving more consistent and adaptable novel view synthesis.

III. METHOD

In this section, we first provide an overview of the proposed UniqueSplat in Section III-A. We then describe the view-conditioned hypernetwork in Section III-B and introduce the primary network in Section III-C.

A. Overview

Figure 2 provides an overview of our proposed method, which consists of a primary network and a view-conditioned hypernetwork. The primary network incorporates an encoder and a predictor, while the hypernetwork is further divided into two complementary branches: view-agnostic and view-specific. This dual structure allows the model to combine both general and distinctive information for multiple views.

Given K sparse-view images $I_c = \{I_c^i\}_{i=1}^K$, their corresponding camera projection matrices $P_c = \{P_c^i\}_{i=1}^K$ and N corresponding camera projection matrices of target views $P_t = \{P_t^i\}_{i=1}^N$, our objective is to render novel views utilizing a feed-forward neural network architecture. The encoder E_p in primary network processes each image to produce a feature volume represented as $F_p = \{F_p^i\}_{i=1}^K$. This encoder captures the intrinsic information within individual views and facilitates the exchange of features across different views. By integrating features from multiple viewpoints, the encoder allows the model to better understand the underlying structure and context of the scene, enhancing the robustness of the extracted representations.

To effectively incorporate view information into our model, we employ a view-conditioned hypernetwork to obtain parameters for the predictor dynamically. Unlike conventional methods that maintain fixed layers during inference, our hypernetwork obtains parameters according to the input images. Specifically, the view-specific branch extracts specific view information by mapping the feature projected from feature

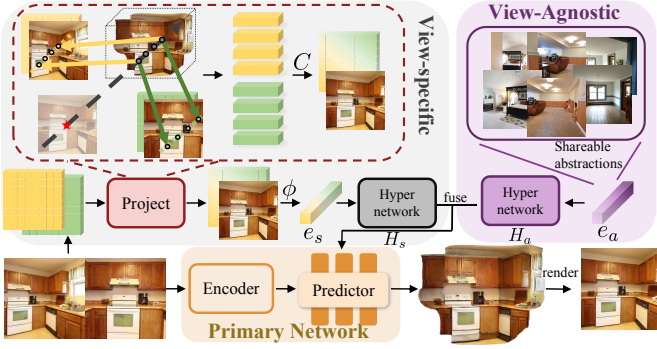


Fig. 2. Overview of our UniqueSplat framework, consisting of two key components: the primary network and the view-conditioned hypernetwork, which includes a view-specific branch and a view-agnostic branch. The view-specific branch leverages the projection matrix to project features from input views (yellow and green) to the target viewpoint, incorporating the knowledge of specific view. It then maps the target view feature into an embedding and calculates a corresponding weight. Meanwhile, the view-agnostic branch extracts shareable knowledge from multiple views of the training scenes and generates another weight via a hypernetwork. Finally, the weights are fused to serve as the weights of the target layers, ensuring the integration of both view-specific and view-agnostic information.

of input views to generate an embedding e_s , which is subsequently processed by the hypernetwork H_s to produce the corresponding weights W_s . Concurrently, learnable view-agnostic embeddings e_a are processed by the view-agnostic hypernetwork H_a to yield weights W_a . We then fuse these weights into weight W_f and apply it to the target layers of the predictor. By incorporating these weights, the predictor in primary network effectively integrates the view prior, allowing it to predict Gaussian parameters including the center μ , opacity $\sum$, covariance σ , and color c . Finally, novel views are rendered from the predicted 3D Gaussians through a splatting operation, enabling the generation of high-fidelity images from various viewpoints.

B. View-conditioned Hypernetwork

In our approach, the view-conditioned hypernetwork injects view priors into the target layers for predicting Gaussian parameters. It consists of two branches: a view-agnostic branch and a view-specific branch.

1) *View-agnostic Branch*: The view-agnostic branch of our model is specifically designed to extract shareable information from diverse input views. This enables the model to maintain a comprehensive understanding of the scene that is not tied to any specific viewpoint.

To achieve this, we introduce learnable embeddings, denoted as e_a , which capture generalizable features which are applicable across multiple viewpoints. We then employ a hypernetwork H_a to project them into a weight matrix W_a as follows:

$$W_a = H_a(e_a), \quad W_a \in \mathbb{R}^{C_{out} \times C_{in}}, \quad (1)$$

where C_{in} and C_{out} denote the input and output dimensions of the target convolution layers respectively. The weight matrix W_a encodes a set of embeddings derived from the diverse

input views of training scenes, which captures information in a manner which is not dependent on a particular viewpoint.

Once the view-agnostic branch generates the weight matrix W_a , we integrate this matrix into the subsequent layers of our primary network. This incorporation allows us to inject the extracted view-agnostic knowledge into the model, allowing the model to leverage this shared information to improve the synthesis quality across different viewpoints.

2) *View-specific Branch*: The view-specific branch is a key component of our model, designed to learn a view-specific embedding vector e_s for each view, which matches the dimensionality of the view-agnostic embedding e_a . Unlike e_a , which remains fixed during testing, the view-specific embedding e_s varies across different views, enabling the network to incorporate the information of specific view for accurate rendering.

To accomplish this, we first employ a backbone encoder E_H to extract per-view features $F_c = \{F_c^i\}_{i=1}^K$ from a series of images $I_c = \{I_c^i\}_{i=1}^K$ where each image I_c^i represents a distinct viewpoint of the same scene. The feature extraction process can be represented as follows:

$$F_c^i = E_H(I_c^i), \quad I_c^i \in \mathbb{R}^{H \times W \times 3}. \quad (2)$$

Following feature extraction, we compute the corresponding projection matrices $P_c = \{P_c^i\}_{i=1}^K$ for each input image, as well as the projection matrices $P_t = \{P_t^i\}_{i=1}^N$ for the target views. Each projection matrix is formulated as:

$$P^i = K^i[R^i|t^i], \quad (3)$$

where K^i represents the intrinsic parameters of the camera, while R^i and t^i denote the rotation and translation matrices, respectively.

To effectively incorporate view prior information, we utilize the target view projection matrix to construct a feature volume by projecting the source features to align with the target view as follows:

$$F_t^i = \text{Proj}(F_c, P_c, P_t^i), \quad (4)$$

where F_c and P_c denote the feature maps and projection matrices of the input views, and P_t^i represents the projection matrix of i -th target view.

For clarity, we derive it in detail, focusing on a single target view as an example. Given the projection matrix $P_t = K_t[R_t|t_t]$ for the target view, we generate a ray for each pixel (u_t, v_t) in the target view. First, we convert the pixel coordinates (u_t, v_t) to normalized camera coordinates by inverting the intrinsic matrix K_t :

$$\mathbf{p}_t = K_t^{-1} \begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix}, \quad (5)$$

where $\mathbf{p}_t = [x_t, y_t, 1]^\top$ represents the point in normalized camera coordinates.

Using the extrinsic parameters $[R_t|t_t]$ of the target view, we map $\mathbf{p}_t$ from the camera coordinate system to the world coordinate system:

$$\mathbf{r} = R_t^\top \mathbf{p}_t, \quad \mathbf{o} = t_t, \quad (6)$$

where $\mathbf{r}$ is the ray direction, and $\mathbf{o}$ is the ray's origin (i.e., the camera's position in world coordinates).

Thus, the ray in world coordinates is represented as:

$$\mathbf{x}(\lambda) = \mathbf{o} + \lambda\mathbf{r}, \quad \lambda \in \mathbb{R}. \quad (7)$$

Next, given the projection matrix $P_c = K_c[R_c|t_c]$ for the source views, we project the ray onto the source views to obtain the corresponding epipolar lines:

$$\mathbf{p}_c(\lambda) = P_c\mathbf{x}(\lambda) = P_c(\mathbf{o} + \lambda\mathbf{r}). \quad (8)$$

This projection results in epipolar lines in the source views, which can be expressed as a linear function of the pixel coordinates $\mathbf{p}_t$:

$$\mathbf{L} = \mathbf{A}\mathbf{p}_t + \mathbf{b}, \quad (9)$$

where $\mathbf{A} = K_c R_c R_t^\top$ represents the relative rotation between the target and source views, and $\mathbf{b} = K_c(R_c t_t + t_c)$ encodes the relative translation. The epipolar lines $\mathbf{L}$ correspond to the projection of the ray onto the image planes of both input views.

Subsequently, we uniformly sample n points along F_c between the near and far planes. We then calculate a correlation matrix to evaluate the relationship between the projected feature from input views for each pixel. We define a function C to compute the correlation matrix of two matrices (M_1, M_2) as follows:

$$C_{ij}(M_1, M_2) = \frac{(M_1^i, M_2^j)}{\|M_1^i\| \|M_2^j\|}, \quad (10)$$

where M_i^j represents the j -th row of matrix M_i , $(\cdot)$ indicates the inner product operation, and $\|\cdot\|$ denotes the norm of a vector. This correlation matrix enables us to effectively fuse the projected features by aggregating them according to their correlation.

Following the fusion of the projected features, we map each target view feature into an embedding that incorporates the specific knowledge of the current view:

$$e_s = \phi(F_t). \quad (11)$$

We then leverage this embedding to obtain the weight matrix W_s through a hypernetwork H_s :

$$\Delta W = H_s(e_s), \quad \Delta W \in \mathbb{R}^{C_{out} \times C_{in}}. \quad (12)$$

Since the dimensions of W_s and W_a are identical, we directly fuse these weights to obtain a weight W which combines both information from distinct branches.

$$W_H = W + \Delta W, \quad W_H \in \mathbb{R}^{C_{out} \times C_{in}}, \quad (13)$$

where W also retains the same dimensions as the target layer. This fusion process enables the model to integrate both general and specific view information, thereby enhancing the overall rendering performance and ensuring high fidelity in the rendered novel views.

C. Primary Network

Similar to MVSplat [1], the primary network consists of an encoder E_p and a predictor. The encoder processes the input images to extract feature volumes $F_p = \{F_p^i\}_{i=1}^K$, where each F_p^i represents the feature map extracted from the input images. This transformation is mathematically described as:

$$F_p = E_p(I_c, P_c), \quad (14)$$

where I_c and P_c represent the input images and their corresponding camera parameters, respectively.

Next, we employ a depth predictor to estimate the parameters of a set of 3D Gaussian primitives including the mean μ , covariance Σ , opacity α , and color c , which collectively define the properties of each primitive in 3D space. In order to ensure that the model effectively incorporates view-conditioned information, we strategically integrate hypernetworks into the convolutional layers of the predictor. This integration allows the model to dynamically adjust based on the specific view query of the scene, thereby improving the accuracy of the predictions.

After obtaining the fused weights via the view-conditioned hypernetwork as in (12), we reshape these weights to match the dimensions required by the convolutional layers. This ensures that the weights are compatible with the architecture of the predictor. These weights are then used by the predictor to generate the Gaussian parameters, which are essential for rendering realistic images from the target viewpoint. The prediction process is formalized as:

$$\mu, \Sigma, \alpha, c = D_{\text{injected}}(F_p), \quad (15)$$

where μ , Σ , α , and c represent the center, covariance, opacity, and color of the Gaussian representation, respectively.

Once the predictor generates the Gaussian parameters, we leverage these predicted Gaussians to render images from the target view. This rendering process is defined as:

$$I_t = R(G, P), \quad (16)$$

where I_t represents the rendered images, G denotes the Gaussians generated by the predictor, and P is the projection matrices of rendered views.

D. Optimization

To train our full model, we utilize ground truth target RGB images to provide supervision during the learning process. In order to ensure a balance between pixel-level accuracy and perceptual quality, we carefully design the training loss as a linear combination of two distinct loss functions: the Mean Squared Error (MSE) loss and the Learned Perceptual Image Patch Similarity (LPIPS) loss. This dual loss approach ensures that the model is capable of capturing both fine-grained pixel-wise details as well as high-level perceptual features. The overall loss is defined as:

$$L = \lambda_1 L_{mse}(\bar{I}_t, I_t) + \lambda_2 L_{lips}(\bar{I}_t, I_t), \quad (17)$$

where $\bar{I}_t$ represents the ground truth target image, and I_t represents the novel view synthesized by the model.

Algorithm 1 UniqueSplat

Input: Input images I_c , camera projection matrix of input views P_c , camera projection matrix of target views P_t

Output: Target views I_t

$W = H_a(e_a)$

for $i \leftarrow 1$ to K **do**

$F_c^i = E_H(I_c^i)$

end for

for $i \leftarrow 1$ to N **do**

$F_t^i = \text{Proj}(F_c, P_c, P_t^i)$

end for

$e_s = \phi(F_t)$

$\Delta W = H_s(e_s)$

$W_H = W + \Delta W$

$F_p = E_p(I_c, P_c)$

$\mu, \Sigma, \alpha, c = D_{\text{injected}}(F_p)$

$I_t = R(G, P_t)$

By optimizing this loss function, our model learns to minimize pixel-wise discrepancies while enhancing perceptual quality, leading to high-fidelity image synthesis.

E. Discussion

1) *Comparison with Ray-Decoupled Image-Based Rendering Methods:* Several prior works in image-based rendering (IBR) adopt strategies that decouple rays, enabling independent ray-based rendering. Specifically, IBRNet [47] employs a ray transformer to aggregate information along each ray within the NeRF decoder, while SRF [44] predicts color and density for each 3D point based on an encoding of its stereo correspondence, implicitly learned from an ensemble of pairwise similarities. Differently, GNT [46] introduces a view transformer to aggregate multi-view image features and a ray transformer to predict target colors, directly regressing pixel colors for view synthesis without relying on NeRF’s volume rendering. GPNR [45] leverages 2D transformer blocks to implicitly capture correspondence by aggregating features along epipolar lines across multiple views, followed by pixel color prediction using the fused features. In contrast, Match-NeRF [64] explicitly computes feature correspondences via cosine similarity for the NeRF conditional inputs. Overall, these methods conduct independent ray-based rendering and do not explicitly model 3D context across rays. In contrast, our method leverages view-conditioned information as a prior, incorporating it into the network to predict explicit pixel-aligned Gaussians for the input views. By explicitly encoding scene geometry, this approach enables more accurate reconstruction of scene structures, as also reflected in the qualitative results.

2) *Comparison with Hypernetwork-Based Novel View Synthesis Methods:* Hypernetworks have recently been applied in novel view synthesis to predict implicit neural representation (INR) weights. For instance, Trans-INR [61] employs a Transformer hypernetwork to infer the entire set of INR weights, removing the single-vector bottleneck and avoiding grid-based representation or gradient computation. HyperINR [62] directly predicts compact INR weights via a hypernet-

work, achieving state-of-the-art inference performance. HyP-NeRF [63] estimates both the weights and multi-resolution hash encodings, yielding quality improvements compared to estimating only the weights of a NeRF. In summary, existing NVS methods primarily focus on directly generating the weights of implicit neural representations from the input signals. In contrast, our approach emphasizes explicit Gaussian representations while incorporating view priors rather than relying solely on the raw input. While prior hypernetwork-based methods typically generate shared weights across different views within the same scene, our method proposes a view-conditioned hypernetwork that predicts distinct weights conditioned on the query view. To the best of our knowledge, UniqueSplat is the first feed-forward Gaussian Splatting model to predict dynamic Gaussians across views. To enable this adaptability, we design a view-conditioned hypernetwork that leverages hypernetworks to adapt across different view queries. Unlike prior approaches that adopt a single-branch design, our hypernetwork consists of two complementary branches: (1) a view-agnostic branch, which captures information shared across diverse views, and (2) a view-specific branch, which captures information unique to each specific view by fusing input-view features with the target view matrix. This dual-branch design is motivated by the observation that scene knowledge can be naturally decomposed into view-specific and view-agnostic components.

IV. EXPERIMENT

In this section, we evaluate the effectiveness of our proposed framework on popular benchmark datasets. We first introduce the datasets, the evaluation metrics for novel view synthesis and implementation details in our experiment. We then present comprehensive experimental results, comparing our method with other state-of-the-art methods both quantitatively and qualitatively. Finally, we perform ablation studies to evaluate the effects of our proposed components in our framework.

A. Settings

Datasets. To comprehensively evaluate the effectiveness of our method, we conduct extensive experiments on two large-scale datasets: RealEstate10K [65] and ACID [66]. The RealEstate10K dataset consists of video frames of real estate scenes, with 67,477 scenes for training and 7,289 scenes for testing. The ACID dataset comprises natural landscape scenes, divided into 11,075 training scenes and 1,972 testing scenes. Both datasets provide estimated camera intrinsic and extrinsic parameters for each frame. Following the setting in pixelSplat [20], we evaluate all methods on three target novel viewpoints per test scene. To further demonstrate the cross-dataset generalization ability of our approach, we additionally evaluate on the multi-view DTU [68] dataset, which consists of object-centric scenes with corresponding camera poses. On DTU dataset, we report results on 16 validation scenes, with 4 novel views for each scene.

Evaluation Metrics. We evaluate the quality of rendered images with three metrics: pixel level PSNR, patch level SSIM [69], and feature-level LPIPS [70]. Higher PSNR and

TABLE I

QUANTITATIVE COMPARISON ON REALSTATE10K [65] AND ACID [66] BENCHMARKS. WE REPORT THE AVERAGE RESULTS OF PSNR, SSIM AND LPIPS ON ALL TEST SCENES. OUR METHOD OUTPERFORMS BASELINES ON ALL EVALUATION METRICS.

Method	RealEstate10K [65]			ACID [66]		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
pixelNeRF [38]	20.43	0.589	0.550	20.97	0.547	0.533
GPNR [45]	24.11	0.793	0.255	25.28	0.764	0.332
AttnRend [48]	24.78	0.820	0.213	26.88	0.799	0.218
MuRF [67]	26.10	0.858	0.143	28.09	0.841	0.155
pixelSplat [20]	25.89	0.858	0.142	28.14	0.839	0.150
MVSplat [1]	26.39	0.869	0.128	28.25	0.843	0.144
TranSplat [18]	26.69	0.875	0.125	28.35	0.845	0.143
Ours	27.28	0.881	0.119	29.06	0.855	0.134

TABLE II

QUALITATIVE COMPARISON FOR CROSS-DATASET GENERALIZATION. ALL MODELS ARE TRAINED ON REALSTATE10K [65] AND TESTED ON ACID [66] AND DTU [68] WITHOUT ANY FINE-TUNING.

Method	ACID [66]			DTU [68]		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
pixelSplat [20]	27.64	0.830	0.160	12.89	0.382	0.560
MVSplat [1]	28.15	0.841	0.147	13.94	0.473	0.385
TranSplat [18]	28.17	0.842	0.146	14.93	0.531	0.326
Ours	28.63	0.848	0.142	16.01	0.665	0.285

SSIM indicate better image quality while lower LPIPS correspond to higher perceptual fidelity. For fair comparison, we follow existing models [1], [18], [20] and conduct all experiments at a resolution of 256×256 .

Architectural Details. For the primary network, the encoder consists of a shallow ResNet-like CNN with 6 residual blocks, followed by a Transformer composed of 6 stacked Transformer blocks. The depth predictor employs a 2D U-Net implementation from [71] to refine the cost volumes, followed by a convolutional block for depth prediction and an additional 2D U-Net for further depth refinement. For the view-specific branch, a ResNet18 network is first applied to extract features from the input views. After fusing these features with the target view matrix, 4 convolutional blocks and a linear layer map the fused features into a view-specific embedding. The view-specific embedding has the same dimensionality as the view-agnostic embedding. Both hypernetworks share the same architecture, a MLP network with 3 layers to map the embedding into the convolutional weights.

Implementation Details. Our implementation is based on the PyTorch framework with an off-the-shelf 3DGS render implemented in CUDA. All experiments are conducted on eight A6000 GPUs, utilizing the Adam optimizer with a learning rate of 1×10^{-6} . To maintain consistency with prior work, the model is trained with two input views and four target views per scene. For all the RealEstate10K, ACID and DTU datasets, all experiments are conducted at a resolution of 256×256 . The encoder and predictor are initialized with those of MVSplat. In the view-conditioned network, the embedding dimension is set to 512. A ResNet backbone is employed

to extract per-view features in the view-specific branch, producing feature maps of size 256×256 . These features are then projected and downsampled to generate the weights for the convolutional layers in the predictor. The training loss is a weighted combination of Mean Squared Error (MSE) and Learned Perceptual Image Patch Similarity (LPIPS) losses. MSE is applied throughout the training process, while LPIPS is introduced after 150,000 iterations. The respective loss weights are set to 1 for MSE and 0.05 for LPIPS.

B. Quantitative Evaluation

In this section, we present quantitative results comparing our method with current state-of-the-art approaches on the RealEstate10K [65], ACID [66] and DTU [68] datasets. We first evaluate the performance of our method against several representative feed-forward methods [38], [45], [48], [67] that focus on novel view synthesis from sparse views in the intra-test setting. GPNR [45] and AttnRend [48] are based on Light Field networks, while pixelNeRF [38] and MuRF [67] are constructed on NeRF. Additionally, we also compare with the state-of-the-art 3DGS-based models [1], [18], [20]. We present quantitative results on the RealEstate10K and ACID datasets in Table I, where our method outperforms all previous state-of-the-art approaches across all metrics, achieving higher rendering quality for both indoor and outdoor scenes. Specifically, our method applies MVSplat [1] as the primary network, and our proposed view-conditioned hypernetwork incorporates view-dependent knowledge, resulting in a 0.89 dB increase in PSNR compared to MVSplat, reaching a PSNR

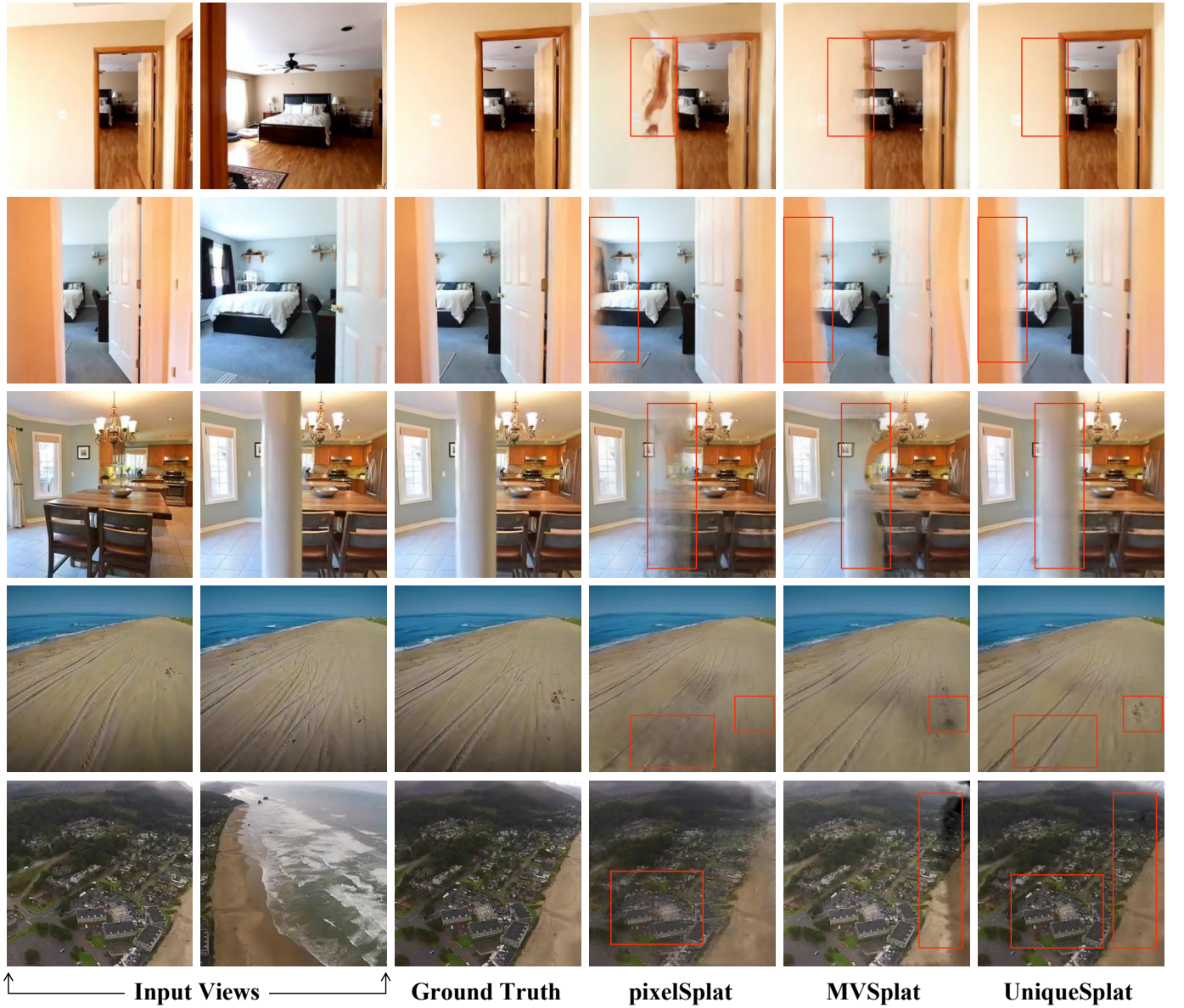


Fig. 3. Qualitative comparison on RealEstate10K [65] and ACID [66] benchmarks. All models are trained and tested on the same datasets. Benefiting from the information of target views, our method outperforms baseline models with higher rendering quality and better details.

of 27.28 dB on the RealEstate10K benchmark. Even when compared to the latest method, TranSplat [18], our approach achieves a notable improvement, surpassing TranSplat by 0.59 dB in PSNR. Furthermore, our method reduces the LPIPS score to 0.119, demonstrating a significant enhancement in perceptual image quality. On the ACID dataset, our method also achieves state-of-the-art results, with a PSNR increase to 29.06 dB and an SSIM of 0.855. The LPIPS score is reduced to 0.134, further highlighting our model’s ability to generate realistic and perceptually consistent views across diverse scenes. To evaluate the generalization ability of UniqueSplat, we conduct cross-dataset experiments, where all methods are trained on the RealEstate10K dataset and tested on other datasets. As shown in Table II, our method achieves superior performance on out-of-distribution novel scenes. This is due to the view-conditioned hypernetwork, which learns

view-dependent information as prior knowledge, enabling the customization of 3D radiance fields. For instance, when trained on the RealEstate10K dataset, our method achieves a PSNR of 28.63 dB, outperforming the current state-of-the-art method, TranSplat [18], by 0.46 dB when tested on the ACID dataset. Notably, our cross-dataset generalization results on ACID even surpass those of MVSplat, which was specifically trained on the ACID dataset. Additionally, we perform experiments in a zero-shot setting on an object-centric dataset. We train our model on the RealEstate10K dataset and directly test it on the DTU dataset. As shown in Table II, UniqueSplat exhibits a significant improvement in generalization compared to previous feed-forward 3D Gaussian Splatting methods, which predict fixed Gaussians. This suggests that incorporating view-conditioned knowledge also enhances generalization ability on object-centric dataset. Our results indicate that our method per-

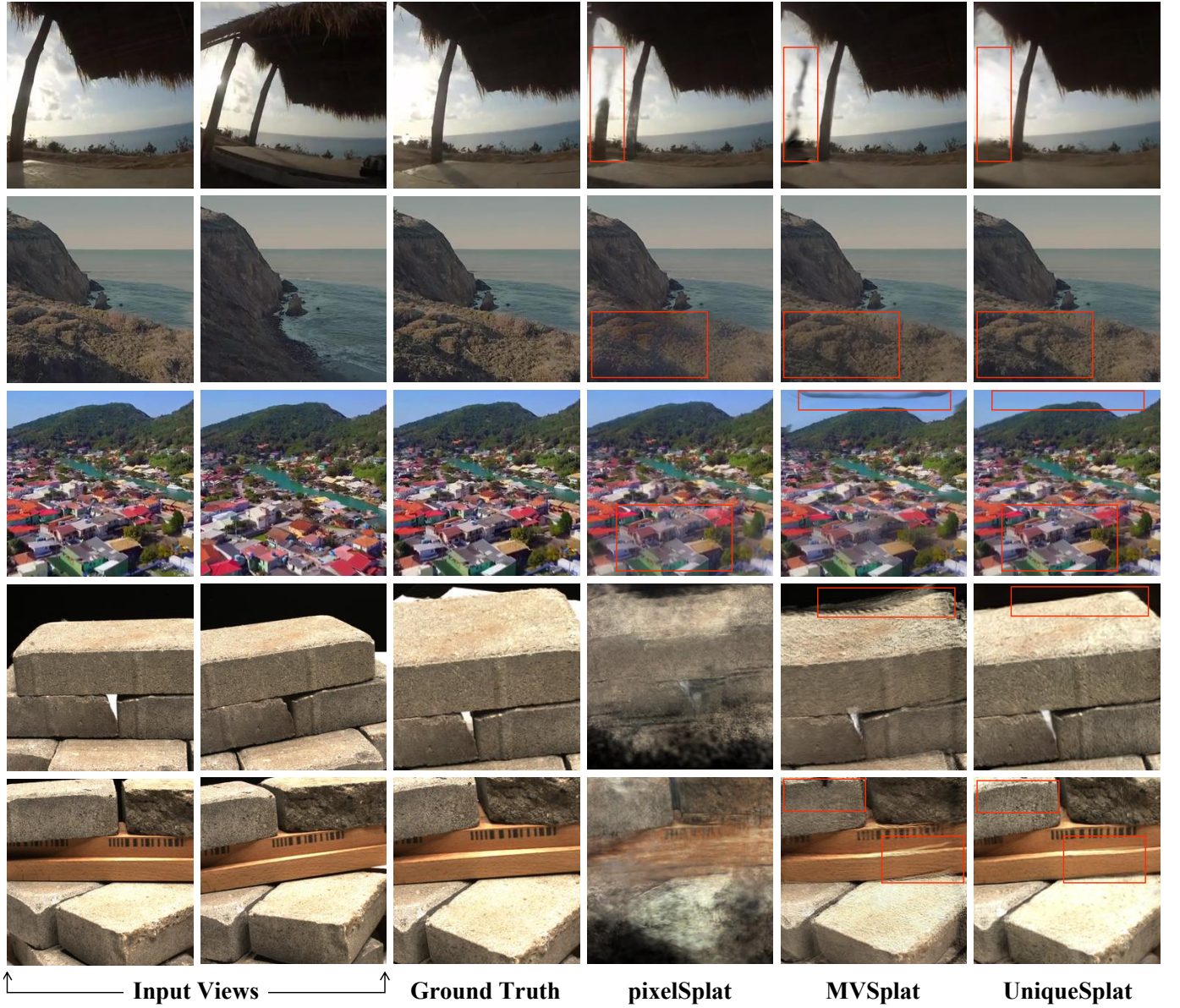


Fig. 4. Qualitative comparison for cross-dataset generalization on ACID [66] and DTU [68]. All models are trained on RealEstate10K [65] and tested on ACID [66] (first three rows) and DTU [68] (last two rows) respectively.

forms well when exposed to views from unseen scenes. While other methods experience a drop in generalization ability due to the domain gap between datasets, our approach is less affected, demonstrating the benefit of the view-conditioned branch for improved generalizability. Overall, our method demonstrates strong performance in both intra-dataset and cross-dataset evaluations, achieving superior results compared with state-of-the-art methods.

C. Qualitative Evaluation

In this section, we present qualitative comparisons with pixelSplat [20] and MVSplat [1]. We first present the intra-test results in Figure 3, with the first three rows corresponding to the RealEstate10K dataset and the last two rows representing the ACID dataset. PixelSplat and MVSplat exhibit cracks and distortions at large viewing angles, while our method

produces high-fidelity, realistic images, benefiting from the view-conditioned information. The first row demonstrates a scene with a large angular transformation, where our method accurately reconstructs the room, while pixelSplat and MVSplat show significant distortions. Additionally, our method excels in reconstructing scene structures, as shown in the third scene, where it successfully reconstructs a column, an area where the other methods struggle. Finally, in outdoor scenes, our method outperforms both pixelSplat and MVSplat, which exhibit blurriness in the last two rows. We further visualize the rendering results in Figure 4, where we train our model on the RealEstate10K dataset and test it on the ACID (first three rows) and DTU (last two rows) datasets. The dataset gap leads to blurry images and artifacts in pixelSplat and MVSplat. In contrast, UniqueSplat leverages additional target view information to render high-quality novel views with less

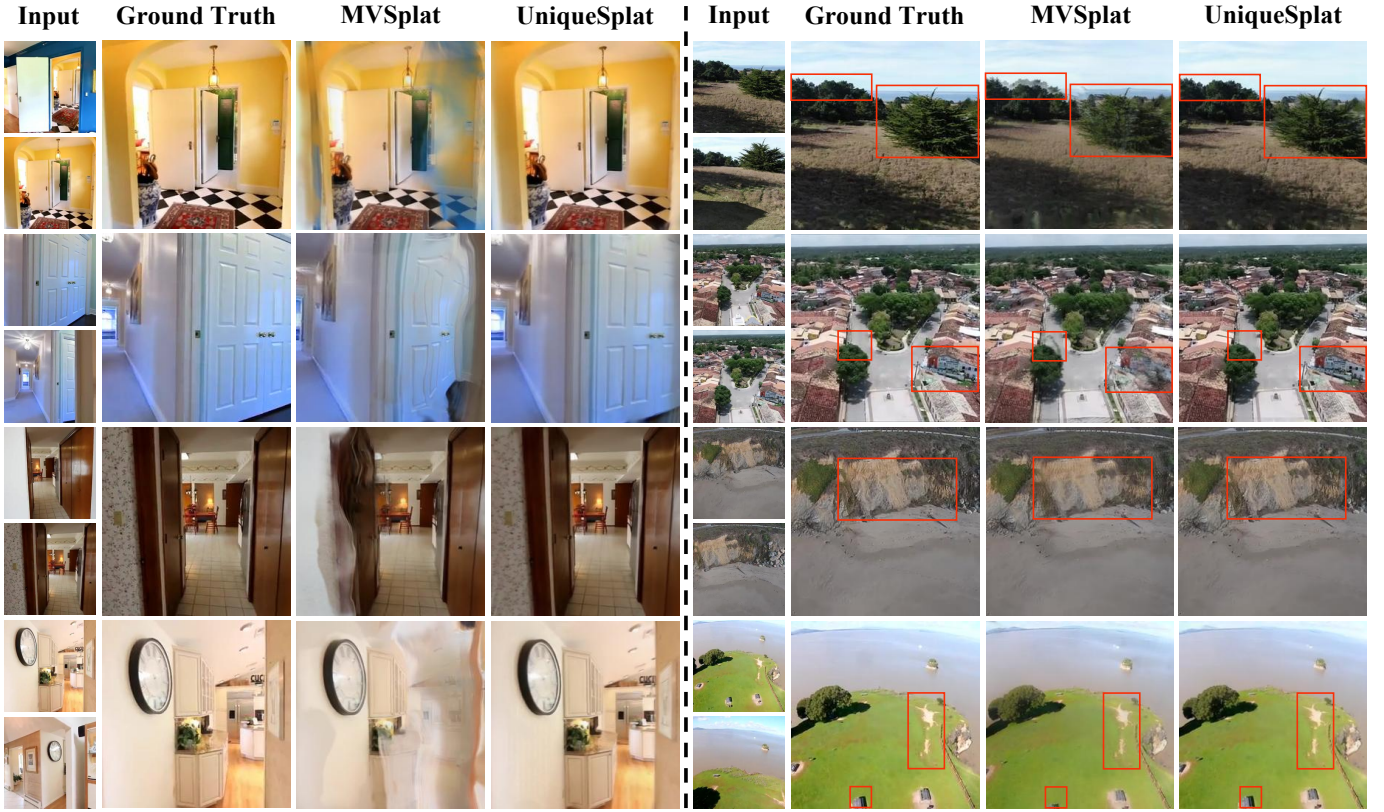


Fig. 5. More qualitative comparisons with MVSpLat. We train models on RealEstate10K dataset and evaluate them on both RealEstate10K and ACID datasets.

edge blurriness, fewer artifacts, and more accurate spatial alignment, demonstrating superior generalization ability. Since MVSpLat serves as our primary network, we provide additional visualizations to demonstrate the effectiveness of incorporating view-conditioned knowledge. We train our models on the RealEstate10K dataset and evaluate them on both the RealEstate10K and ACID datasets. As illustrated in Figure 5, our proposed UniqueSpLat synthesizes novel view with better structure preservation, more intricate details and fewer blurs.

D. Efficiency Analysis

In this section, we provide a detailed comparison of our method with existing approaches in terms of computational resources required for inference. Specifically, we conduct a comprehensive evaluation of all methods on an A6000 GPU, reporting the number of parameters, inference time, memory usage, and PSNR performance on the Re10K, ACID, and DTU datasets.

As shown in Table III, PixelSpLat contains 125.4M parameters, with an inference time of 0.282s and 4.19 GB of memory usage. These costs are primarily due to its epipolar transformer. MVSpLat leverages a cost-volume encoder, achieving superior quantitative results with a lightweight model size and fast speed. In particular, MVSpLat requires only 12.0M parameters, an inference time of 0.048s, and 1.68 GB memory usage, while outperforming PixelSpLat by 0.5, 0.11, and 1.05 in PSNR on the three datasets, respectively.

TransSpLat [18] builds upon MVSpLat by introducing a transformer-based module to produce high-confidence depth

estimates and incorporating monocular depth priors. While this design provides modest PSNR improvements, it comes with higher computational costs, including 110.5M parameters, an inference time of 0.083s, and 4.98 GB memory.

In contrast, our method also extends MVSpLat but leverages a two-branch view-conditioned hypernetwork to reconstruct customized Gaussians for each query view. This design achieves superior PSNR performance (27.28, 29.06, and 16.01 on Re10K, ACID, and DTU, respectively) while requiring lower computational resources (38.1M parameters, an inference time of 0.065s, and 3.12 GB memory) compared with TransSpLat, demonstrating a more efficient trade-off between performance and cost.

E. Ablation Study

In this section, we conduct a series of ablation studies to evaluate the effectiveness of the proposed view-conditioned hypernetworks in our framework, as shown in Table IV. We start by introducing variant (a), a baseline model that follows a standard view synthesis pipeline. This model consists of a primary network, which includes a backbone encoder and a predictor, without additional view-conditioned components. Next, we introduce variant (b), which incorporates a view-agnostic branch (VA) to abstract general scene knowledge across various views from multiple scenes. This branch aims to capture shared features that are not specific to any single view, thus contributing to a more robust and stable training process. Benefiting from the shareable information across diverse views, the model retains a comprehensive understanding

TABLE III

COMPARISON OF COMPUTATIONAL RESOURCES AMONG DIFFERENT FEED-FORWARD GAUSSIAN SPLATTING METHODS. WE REPORT THE NUMBER OF PARAMETERS (M), INFERENCE TIME (S), AND MEMORY USAGE (GB), ALONG WITH PSNR PERFORMANCE ON THE RE10K [65], ACID [66], AND DTU [68] DATASETS.

Method	Params (M)	Time (s)	Memory (GB)	PSNR		
				Re10K [65]	ACID [66]	DTU [68]
pixelSplat [20]	125.4	0.282	4.19	25.89	28.14	12.89
MVSplat [1]	12.0	0.048	1.68	26.39	28.25	13.94
TransSplat [18]	110.5	0.083	4.98	26.69	28.35	14.93
Ours	38.1	0.065	3.12	27.28	29.06	16.01

TABLE IV

ABLATION STUDIES ON THE REALESTATE10K [65] AND ACID [66] DATASETS. VA REPRESENTS OUR VIEW-AGNOSTIC BRANCH AND VS IS OUR VIEW-SPECIFIC BRANCH. WE REPORT INFERENCE TIME AS WELL AS THE AVERAGE PSNR, SSIM, AND LPIPS ON ALL TEST SCENES.

ID	VA	VS	Time (ms)	RealEstate10K [65]			ACID [66]		
				PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
(a)	-	-	48.5	26.39	0.869	0.128	28.25	0.843	0.144
(b)	✓	-	48.6	26.70	0.873	0.124	28.70	0.851	0.139
Ours	✓	✓	65.2	27.28	0.881	0.119	29.06	0.855	0.134

of scenes, leading to an increase of 0.31dB and 0.45dB on PSNR, compared to the vanilla model on RealEstate10K and ACID benchmarks, respectively. The view-agnostic branch introduces only 0.1 ms additional latency, since it employs an MLP module to generate weights. Our view-specific branch is proposed to dynamically adjust the predictor to provide the additional geometry information for the target views, via combining features of input views for each target view query to extract view-specific knowledge. By integrating both view-specific and view-agnostic knowledge, our UniqueSplat achieves the best PSNR of 27.28dB and 29.06dB on both RealEstate10K and ACID benchmarks. The view-specific branch introduces approximately 16.6 ms additional latency, as it relies on a ResNet18 backbone to extract features from two input views and then projects them into the target view matrix. This projection step is the primary contributor to the increased inference latency. The quantitative results demonstrate the efficacy of the proposed view-conditioned hypernetwork. The integration of both the view-agnostic and view-specific branches leads to superior performance across multiple benchmarks, confirming the importance of incorporating both types of scene knowledge for robust and high-quality view synthesis.

F. Analysis

- Our UniqueSplat demonstrates strong performance on widely-used datasets. Unlike existing feed-forward 3D Gaussian Splatting models, which predict fixed Gaussians for each scene, UniqueSplat customizes Gaussians for each view query by incorporating view-conditioned knowledge.
- The proposed view-conditioned hypernetwork injects view-conditioned knowledge into the primary network via generating the parameters of primary network. Compared



Fig. 6. Failure case. While UniqueSplat delivers impressive results, it fails to accurately reconstruct unseen regions

to other state-of-the-art methods, the design of view-conditioned hypernetwork achieves superior performance in intra-test and cross-test settings. Qualitative results further demonstrate that our method synthesizes high-quality images with fewer cracks, blurs and artifacts.

- Evaluations of both the view-agnostic and view-specific branches in the view-conditioned hypernetwork show that the introduction of both branches encouragingly improves the performance of novel view synthesis. The view-agnostic branch abstracts the characteristics shared across various views of the training scenes, contributing to a robust and stable training process. The view-specific branch introduces multi-view priors, dynamically adjusting the model to accommodate each specific view.

V. LIMITATIONS AND FUTURE WORK

While UniqueSplat achieves compelling results and outperforms previous methods, it still has certain limitations. Specifically, because it predicts pixel-aligned Gaussians for each input view, UniqueSplat may introduce artifacts when rendering unseen regions, as shown in Figure 6. Future work

could address these issues by introducing generative models, such as diffusion models.

VI. CONCLUSION

In this paper, we have proposed UniqueSplat, a novel view-conditioned 3D Gaussian Splatting model that reconstructs customized 3D radiance fields for each view query. Conventional feed-forward methods directly utilize fixed Gaussians to render images across all views, struggling to specialize to specific view. To address this, UniqueSplat dynamically adjusts Gaussians in accordance with different views by learning the view-conditioned knowledge as a prior. It simultaneously learns the view-agnostic embeddings and view-specific knowledge, incorporating shareable knowledge from various view of 3D scenes and adapting the Gaussians dynamically according to the given views. Extensive experiments on benchmark datasets, including RealEstate10K, ACID and DTU, demonstrate that UniqueSplat outperforms state-of-the-art methods, both in terms of image quality and generalization ability.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 62576185, and by the Beijing Natural Science Foundation under Grant L252011.

REFERENCES

- [1] Y. Chen, H. Xu, C. Zheng, B. Zhuang, M. Pollefeys, A. Geiger, T.-J. Cham, and J. Cai, “Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images,” in *European Conference on Computer Vision*. Springer, 2024, pp. 370–386.
- [2] S. Zheng, B. Zhou, R. Shao, B. Liu, S. Zhang, L. Nie, and Y. Liu, “Gps-gaussian: Generalizable pixel-wise 3d gaussian splatting for real-time human novel view synthesis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 680–19 690.
- [3] C. Jiang, H. Zhang, P. Liu, Z. Yu, H. Cheng, B. Zhou, and S. Shen, “ H_2 -mapping: Real-time dense mapping using hierarchical hybrid representation,” *IEEE Robotics and Automation Letters*, 2023.
- [4] A. Tonderski, C. Lindström, G. Hess, W. Ljungbergh, L. Svensson, and C. Petersson, “Neurad: Neural rendering for autonomous driving,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 895–14 904.
- [5] Y. Wang, Q. Zhao, Y. Gan, and Z. Xia, “Joint-confidence-guided multi-task learning for 3d reconstruction and understanding from monocular camera,” *IEEE Transactions on Image Processing*, vol. 32, pp. 1120–1133, 2023.
- [6] H. Cui, D. Tu, F. Tang, P. Xu, H. Liu, and S. Shen, “Vidsfm: Robust and accurate structure-from-motion for monocular videos,” *IEEE Transactions on Image Processing*, vol. 31, pp. 2449–2462, 2022.
- [7] J. Li, Z. Lu, Y. Wang, J. Xiao, and Y. Wang, “Nr-mvsnet: Learning multi-view stereo based on normal consistency and depth refinement,” *IEEE Transactions on Image Processing*, vol. 32, pp. 2649–2662, 2023.
- [8] J. Jin and J. Hou, “Occlusion-aware unsupervised learning of depth from 4-d light fields,” *IEEE Transactions on Image Processing*, vol. 31, pp. 2216–2228, 2022.
- [9] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, “Neural sparse voxel fields,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 651–15 663, 2020.
- [10] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, “Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5855–5864.
- [11] 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.
- [12] T. Müller, A. Evans, C. Schied, and A. Keller, “Instant neural graphics primitives with a multiresolution hash encoding,” *ACM Transactions on Graphics*, vol. 41, no. 4, pp. 1–15, 2022.
- [13] M. Chen, L. Wang, Y. Lei, Z. Dong, and Y. Guo, “Learning spherical radiance field for efficient 360 unbounded novel view synthesis,” *IEEE Transactions on Image Processing*, 2024.
- [14] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering,” *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 1–14, 2023.
- [15] Z. Yan, W. F. Low, Y. Chen, and G. H. Lee, “Multi-scale 3d gaussian splatting for anti-aliased rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 923–20 931.
- [16] Z. Yu, A. Chen, B. Huang, T. Sattler, and A. Geiger, “Mip-splatting: Alias-free 3d gaussian splatting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 447–19 456.
- [17] S. Szymanowicz, C. Rupprecht, and A. Vedaldi, “Splatter image: Ultra-fast single-view 3d reconstruction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 208–10 217.
- [18] C. Zhang, Y. Zou, Z. Li, M. Yi, and H. Wang, “Transplat: Generalizable 3d gaussian splatting from sparse multi-view images with transformers,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 9, 2025, pp. 9869–9877.
- [19] S. Zhang, X. Fei, F. Liu, H. Song, and Y. Duan, “Gaussian graph network: Learning efficient and generalizable gaussian representations from multi-view images,” *Advances in Neural Information Processing Systems*, 2024.
- [20] D. Charatan, S. L. Li, A. Tagliasacchi, and V. Sitzmann, “pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 457–19 467.
- [21] 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,” in *European Conference on Computer Vision*. Springer, 2020, pp. 405–421.
- [22] 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.
- [23] W. Zhang, R. Xing, Y. Zeng, Y.-S. Liu, K. Shi, and Z. Han, “Fast learning radiance fields by shooting much fewer rays,” *IEEE Transactions on Image Processing*, vol. 32, pp. 2703–2718, 2023.
- [24] Y. Jiang, J. Tu, Y. Liu, X. Gao, X. Long, W. Wang, and Y. Ma, “Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5322–5332.
- [25] Z. Liang, Q. Zhang, Y. Feng, Y. Shan, and K. Jia, “Gs-ir: 3d gaussian splatting for inverse rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 644–21 653.
- [26] T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin, and B. Dai, “Scaffold-gs: Structured 3d gaussians for view-adaptive rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 654–20 664.
- [27] K. Navaneet, K. P. Meibodi, S. A. Koohpayegani, and H. Pirsiavash, “CompGs: Smaller and faster gaussian splatting with vector quantization,” *European Conference on Computer Vision*, 2024.
- [28] S. Girish, K. Gupta, and A. Shrivastava, “Eagles: Efficient accelerated 3d gaussians with lightweight encodings,” *European Conference on Computer Vision*, 2024.
- [29] 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*. Springer, 2024, pp. 145–163.
- [30] H. Xiong, S. Muttukuru, R. Upadhyay, P. Chari, and A. Kadambi, “Sparsegs: Real-time 360° sparse view synthesis using gaussian splatting,” *arXiv preprint arXiv:2312.00206*, 2023.
- [31] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, “4d gaussian splatting for real-time dynamic scene rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 310–20 320.
- [32] Z. Yang, H. Yang, Z. Pan, and L. Zhang, “Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting,” in *The Twelfth International Conference on Learning Representations*.
- [33] T. Xie, Z. Zong, Y. Qiu, X. Li, Y. Feng, Y. Yang, and C. Jiang, “Phys-gaussian: Physics-integrated 3d gaussians for generative dynamics,” in

- Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4389–4398.
- [34] X. Liu, C. Zhou, and S. Huang, “3dgs-enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 133 305–133 327, 2024.
- [35] W. Xu, H. Gao, S. Shen, R. Peng, J. Jiao, and R. Wang, “Mvpgs: Excavating multi-view priors for gaussian splatting from sparse input views,” in *European Conference on Computer Vision*. Springer, 2024, pp. 203–220.
- [36] H. Zhou, W. Dong, and J. Chen, “Lita-gs: Illumination-agnostic novel view synthesis via reference-free 3d gaussian splatting and physical priors,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025, pp. 21 580–21 589.
- [37] J. Chung, J. Oh, and K. M. Lee, “Depth-regularized optimization for 3d gaussian splatting in few-shot images,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2024, pp. 811–820.
- [38] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, “pixelnerf: Neural radiance fields from one or few images,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4578–4587.
- [39] W. Jang and L. Agapito, “Codenerf: Disentangled neural radiance fields for object categories,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 949–12 958.
- [40] K. Rematas, R. Martin-Brualla, and V. Ferrari, “Sharf: Shape-conditioned radiance fields from a single view,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 8948–8958.
- [41] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su, “Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 124–14 133.
- [42] M. M. Johari, Y. Lepoittevin, and F. Fleuret, “Geonerf: Generalizing nerf with geometry priors,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 365–18 375.
- [43] Y. Liu, S. Peng, L. Liu, Q. Wang, P. Wang, C. Theobalt, X. Zhou, and W. Wang, “Neural rays for occlusion-aware image-based rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7824–7833.
- [44] J. Chibane, A. Bansal, V. Lazova, and G. Pons-Moll, “Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7911–7920.
- [45] M. Suhail, C. Esteves, L. Sigal, and A. Makadia, “Generalizable patch-based neural rendering,” in *European Conference on Computer Vision*. Springer, 2022, pp. 156–174.
- [46] M. Varma, P. Wang, X. Chen, T. Chen, S. Venugopalan, and Z. Wang, “Is attention all that nerf needs?” in *The Eleventh International Conference on Learning Representations*.
- [47] Q. Wang, Z. Wang, K. Genova, P. P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, and T. Funkhouser, “Ibrnet: Learning multi-view image-based rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4690–4699.
- [48] Y. Du, C. Smith, A. Tewari, and V. Sitzmann, “Learning to render novel views from wide-baseline stereo pairs,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4970–4980.
- [49] 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.
- [50] C. Sun, M. Sun, and H.-T. Chen, “Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5459–5469.
- [51] S. Szymanowicz, C. Rupprecht, and A. Vedaldi, “Viewset diffusion:(0-) image-conditioned 3d generative models from 2d data,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8863–8873.
- [52] T. Anciukevičius, Z. Xu, M. Fisher, P. Henderson, H. Bilen, N. J. Mitra, and P. Guerrero, “Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 608–12 618.
- [53] J. Gu, A. Trevithick, K.-E. Lin, J. M. Susskind, C. Theobalt, L. Liu, and R. Ramamoorthi, “Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 11 808–11 826.
- [54] C. Wewer, K. Raj, E. Ilg, B. Schiele, and J. E. Lenssen, “latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction,” in *European Conference on Computer Vision*. Springer, 2024, pp. 456–473.
- [55] D. Ha, A. Dai, and Q. V. Le, “Hypernetworks,” *arXiv preprint arXiv:1609.09106*, 2016.
- [56] V. K. Chauhan, J. Zhou, P. Lu, S. Molaei, and D. A. Clifton, “A brief review of hypernetworks in deep learning,” *Artificial Intelligence Review*, vol. 57, no. 9, p. 250, 2024.
- [57] Y. Nirkin, L. Wolf, and T. Hassner, “Hyperseg: Patch-wise hypernetwork for real-time semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4061–4070.
- [58] A. Brock, T. Lim, J. Ritchie, and N. Weston, “Smash: One-shot model architecture search through hypernetworks,” in *International Conference on Learning Representations*, 2018.
- [59] F. Szatkowski, K. J. Piczak, P. Spurek, J. Tabor, and T. Trzcinski, “Hypersound: Generating implicit neural representations of audio signals with hypernetworks,” *arXiv preprint arXiv:2211.01839*, 2022.
- [60] C. Gordon, L. E. MacDonald, H. Saratchandran, and S. Lucey, “D’oh: Decoder-only random hypernetworks for implicit neural representations,” in *Proceedings of the Asian Conference on Computer Vision*, 2024, pp. 2507–2526.
- [61] Y. Chen and X. Wang, “Transformers as meta-learners for implicit neural representations,” in *European Conference on Computer Vision*. Springer, 2022, pp. 170–187.
- [62] Q. Wu, D. Bauer, Y. Chen, and K.-L. Ma, “Hyperinr: A fast and predictive hypernetwork for implicit neural representations via knowledge distillation,” *arXiv preprint arXiv:2304.04188*, 2023.
- [63] B. Sen, G. Singh, A. Agarwal, R. Agaram, M. Krishna, and S. Sridhar, “Hyp-nerf: Learning improved nerf priors using a hypernetwork,” *Advances in Neural Information Processing Systems*, vol. 36, 2023.
- [64] Y. Chen, H. Xu, Q. Wu, C. Zheng, T.-J. Cham, and J. Cai, “Explicit correspondence matching for generalizable neural radiance fields,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.
- [65] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, “Stereo magnification: Learning view synthesis using multiplane images,” *arXiv preprint arXiv:1805.09817*, 2018.
- [66] 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 International Conference on Computer Vision*, 2021, pp. 14 458–14 467.
- [67] 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. 20 041–20 050.
- [68] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanaes, “Large scale multi-view stereopsis evaluation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014, pp. 406–413.
- [69] 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.
- [70] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595.
- [71] 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. 10 684–10 695.



Haixu Song received the B.Eng. degree in Automation and the M.Eng. degree in Control Science and Engineering from the Harbin Institute of Technology in 2020 and 2022, respectively. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, Tsinghua University. His main interest is 3D computer vision.



Xiaoke Yang received the B.Eng. degree from the School of Computer Science, Beijing University of Posts and Telecommunications, in 2022. He is currently pursuing the M.Eng. degree with the Tsinghua Shenzhen International School, Tsinghua University. His research interests include 3D deep learning and Deepfake Detection.



Shengjun Zhang received the B.Eng degree from the Department of Engineering Physics of Tsinghua University, Beijing, China, in 2023, where he is currently pursuing the Ph.D degree with the Department of Electronic Engineering. His main research interest is 3D computer vision.



Jiwen Lu (Fellow, IEEE) received the B.Eng. degree in mechanical engineering and the M.Eng. degree in electrical engineering from the Xi'an University of Technology, Xi'an, China, in 2003 and 2006, respectively, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2012. He is currently a Professor with the Department of Automation, Tsinghua University, Beijing, China. His current research interests include computer vision and pattern recognition, where he has authored/coauthored more than 160 scientific

articles in IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, International Journal of Computer Vision, CVPR, ICCV, and ECCV. He serves as the Co-Editor-in-Chief for Pattern Recognition Letters and an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON BIOMETRICS, BEHAVIOR, IDENTITY SCIENCE, and Pattern Recognition. He also serves as the Program Co-Chair for FG'2023, VCIP'2022, AVSS'2021, and ICME'2020. He is a recipient of the National Outstanding Youth Foundation of China Award and an IAPR Fellow.



Yueqi Duan (Member, IEEE) received the B.S. and Ph.D. degrees from the Department of Automation, Tsinghua University, in 2014 and 2019, respectively. From 2019 to 2021, he was a Post-Doctoral Researcher with the Computer Science Department, Stanford University. He is currently an Assistant Professor with the Department of Electronic Engineering, Tsinghua University. He has published more than 30 scientific papers in the top journals and conferences, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, CVPR, ICCV, ECCV and NeurIPS. His research interests include computer vision and pattern recognition. He served as the Publication Chair for FG, the Area Chair for CVPR, ICLR, MM and ICME, and a Regular Reviewer for a number of journals and conferences, e.g., IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, IJCV, CVPR, ICCV, ECCV, ICML, NeurIPS, and SIGGRAPH. He was awarded the Excellent Doctoral Dissertation of Chinese Association for Artificial Intelligence (CAAI) in 2020.