Pointmap-Conditioned Diffusion for Consistent Novel View Synthesis

Thang-Anh-Quan Nguyen[1] Laurent Caraffa[2] Jean-Philippe Tarel[1] Roland Brémond[1]
[1]Gustave Eiffel University, France
[2]IGN-ENSG, France



Figure 1. PointmapDiff is a method that can perform extrapolated view synthesis in urban scenes. We present viewpoints generated at 45° angle to the right (first row) and at $1.5m$ position to the left (second row). Our approach significantly outperforms the baselines when rendering viewpoints beyond the original recorded trajectory, whereas 3DGS [13] struggles with severe artifacts.

Abstract

*Synthesizing extrapolated views remains a difficult task, especially in urban driving scenes, where the only reliable sources of data are limited RGB captures and sparse LiDAR points. To address this problem, we present PointmapDiff, a framework for novel view synthesis that utilizes pre-trained 2D diffusion models. Our method leverages point maps (*i.e.*, rasterized 3D scene coordinates) as a conditioning signal, capturing geometric and photometric priors from the reference images to guide the image generation process. With the proposed reference attention layers and ControlNet for point map features, PointmapDiff can generate accurate and consistent results across varying viewpoints while respecting geometric fidelity. Experiments on real-life driving data demonstrate that our method achieves high-quality generation with flexibility over point map conditioning signals (*e.g.*, dense depth map or even sparse LiDAR points) and can be used to distill to 3D representations such as 3D Gaussian Splatting for improving view extrapolation.*

1. Introduction

Reconstruction of urban driving scenes plays a crucial role in understanding and advancing autonomous driving systems. Recently, neural rendering techniques such as Neural Radiance Fields (NeRFs) [18, 20, 21, 49, 52] and 3D Gaussian Splatting (3DGS) [13, 51, 63] have demonstrated remarkable potential in synthesizing photorealistic street views, allowing autonomous vehicles to be trained and tested in more diverse and complex scenarios.

Despite these advancements, a significant challenge persists in the form of extrapolation, where the model struggles to render images from viewpoints that differ significantly from the recorded data. Since most training images are captured from vehicle-mounted cameras with low overlap on a single trajectory, the neural reconstruction models primarily learn to interpolate on this trajectory rather than extrapolate to far-away views. This results in a degraded rendering quality, with noticeable blur and artifacts when synthesizing views from extreme angles and positions. Addressing this limitation is crucial for maximizing the utility of reconstructed street scenes, ensuring autonomous driving simulations remain accurate even when operating in regions that are not directly captured during training.

To this extent, we introduce PointmapDiff, a framework designed to leverage pre-trained 2D diffusion models for novel view synthesis (NVS) by incorporating 3D structure into 2D diffusion features. In urban driving environments, LiDAR provides reliable geometric information with broader coverage, whereas relying solely on RGB images is insufficient for capturing the scene. To address this, PointmapDiff leverages ControlNet conditioning on point

maps, 2D projections of 3D coordinates from the scene's point cloud, and combines them with features extracted from reference images. This conditioning allows the model to capture relevant geometric relationships between viewpoints. Additionally, PointmapDiff employs a reference cross-view attention module, which ensures implicit transfer of information from reference views to the generated target views.

Our approach offers two key benefits: first, point maps establish better correspondences between viewpoints compared to RGB, which require texture details and constant lighting; and second, the model can adapt point maps derived from sparse LiDAR data as geometric guidance for generation, in scenarios where establishing correspondences is particularly challenging. Finally, we show that PointmapDiff can be used to generate views that are aligned with LiDAR scans. This also serves as supplementary supervision for refining scene representation, such as 3DGS, beyond the initial training trajectories. By integrating the benefits of distilling 3D scene modeling with diffusion-based image generation, PointmapDiff achieves state-of-the-art performance in extrapolated-view synthesis, delivering high-quality, consistent renderings of unobserved views. To summarize, our main contributions are as follows:

- we propose a point map-conditioned generative framework that can synthesize viewpoints from a single or multiple reference views,
- by effectively capturing features and correspondences from point maps, our results consistently respect both geometric and appearance information from reference views and LiDAR scans,
- we showcase PointmapDiff's performance in urban reconstruction, given restrictions in sparse points input, as well as its effectiveness in single-image NVS and object manipulation.

2. Related Work

Novel View Synthesis. The goal of NVS is to generate realistic and visually coherent images of a specific instance or scene from camera viewpoints that have not been observed before. This involves taking one or more existing views of the scene and synthesizing new views while ensuring consistency. NVS can be categorized into two types based on how viewpoints are generated: *View Interpolation*, where the synthesized viewpoints lie within the given input views distribution, and *View Extrapolation*, which involves generating viewpoints outside the input range, often requiring the model to infer a larger amount of content.

Many modern view interpolation methods are *reconstruction-based* and built upon NeRF [18], 3GDS [13], and their derivatives [39], which describe a scene as radiance fields to fit the observed images. They enable 3D representation by capturing photos of a real scene and optimizing the underlying geometry and appearance. However, these methods typically require extensive per-scene fitting and only allow for rendering the scene from viewpoints in the training pose distribution. As a result, they usually struggle to generate realistic details in far-away viewpoints. Moreover, capturing detailed scenes requires hundreds to thousands of images, while insufficient scene coverage can lead to optimization issues, resulting in inaccurate geometry, artifacts, and blurry renderings.

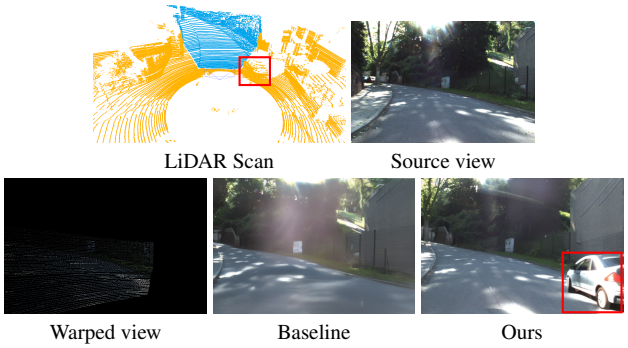


Figure 2. From a reference image and synchronized LiDAR scan, while the image can observe only a small part (blue) of the scene, the geometric information from the rest of the LiDAR scan (orange) can still be used to generate meaningful content. We label the cars that appear in both the LiDAR scan and the generated image in red, denoting the advantage of our method compared to other baselines.

On the other hand, most existing extrapolation methods are *generative-based* and rely on training generative models to take available reference images and camera viewpoints as conditions, and directly generate new views. These methods are designed to work with minimal input (*e.g.*, a single image) and rely on general knowledge from large models and datasets. ReconFusion [48] use priors from CLIP image embedding [22] and pixelNeRF's [54] features for enabling 3D-awareness. Other works [7, 24, 38, 40, 55, 56] designed special attention mechanisms based on epipolar geometry, local neighborhoods, or camera's ray embeddings [32]. GenWarp [30], MultiDiff [19], and ViewCrafter [57] focus on implicit geometric warping signals using Monocular Depth Estimation (MDE) [2, 23]. Still, closing the domain gap between indoor and outdoor scenes remains challenging as MDE becomes less accurate, while sparse LiDAR data makes it difficult to obtain sufficient warping information.

Extrapolation in Street View Reconstruction. In driving scenes, the training camera distribution is often biased towards forward-facing movements, which severely limits the vehicle's field of view. Additionally, these straight trajectories could lead to overfitting in methods that rely only on camera parameters. To improve the rendering quality of neural rendering models for viewpoints distant from the

training views, several methods augment training with synthesized viewpoints from external generative models. For instance, VEGS [12] employs a diffusion model with per-scene LoRA [10] and Perturb-and-Average Scoring [42] to enhance details in extrapolated views. However, this approach lacks control over geometry and thus depends heavily on strong geometric priors, such as normal supervision. Without such priors, it would require longer training times for the distillation loss to converge. Similarly, SGD [58] trains a diffusion model with a ControlNet conditioned on two adjacent frames and the dense depth prediction of the current frame, focusing on few-shot setups. Yet, it uses patchified CLIP image embeddings [22] as guidance, which provide high-level semantic information but lack precise spatial detail. This results in inconsistencies in the generated images that can negatively impact the overall 3DGS training. In contrast, our approach focuses on improving the quality of the generative model, enabling more accurate and stable results. We believe that a reliable model with better adaptation to the data modalities commonly found in driving scenarios can significantly boost the performance. FreeVS [44] utilizes colorized LiDAR point clouds to generate pseudo-images for conditioning, making it the most comparable approach to ours. However, like many other baselines, it frequently struggles in regions with absent RGB coverage (Fig. 2). This limitation presents a significant challenge to the safety and reliability of autonomous perception. Moreover, instead of training the entire architecture across large datasets with numerous parameters, we design our method to optimize only a minimal subset of parameters by leveraging ControlNet [59]. We demonstrate that it eliminates the need for extensive fine-tuning and still achieves satisfactory results.

3. Method

3.1. Preliminaries

Diffusion Models [9] are probabilistic models designed to learn the underlying data distribution $p(x)$ by starting from a Gaussian distributed variable x_T and gradually denoising it to recover the original data sample x_0, which simulates the reverse process of a fixed forward (noise-adding) Markov Chain.

In particular, we leverage Latent Diffusion Models (LDM) [27], which utilize a pre-trained Variational Auto-Encoder (VAE) [14] to map image data from pixel space into a compressed latent space with lower dimensionality and perform diffusion process in that latent space. This reduces computational complexity, memory footprint, and enables conditioning on other modalities, such as text during generation, while still preserving details. Typically, to learn the denoising process, the network, U-Net [28] in this case, is trained to predict the noise by minimizing:

$$\mathcal{L}(\theta) = \mathbb{E}_{\epsilon,\tau}\left[\|\epsilon_\theta(z_\tau, \tau, \mathbf{c}) - \epsilon\|_2^2\right], \tag{1}$$

where ϵ_θ is the noise prediction network with parameters θ, $\tau \sim \mathcal{U}(0, T)$ is the time step, z_τ is the noisy latent at τ, $\epsilon \sim \mathcal{N}(0, I)$ is the additive Gaussian noise, and $\mathbf{c}$ denotes the user-specified condition signal, which is used for the conditional generation.

ControlNet [59] is a versatile network that allows the addition of conditioning into a pre-trained Stable Diffusion (SD) model. It has been demonstrated to support various types of input conditioning, such as depth, sketches, and semantic maps, by injecting conditional image features into trainable copies of the original SD encoder blocks, allowing SD to generate images that are coherent with the input condition. A key advantage of ControlNet is its ability to resist overfitting during fine-tuning, allowing it to retain the original model's performance. This makes it particularly useful for incorporating 3D awareness [43, 50] into diffusion models without compromising their 2D semantic quality. Nonetheless, this ability has not yet been fully exploited in view synthesis applications.

Problem Statement. Given a reference RGB image I^r with its geometric cue D^r (derived from a depth sensor, estimated depth map or LiDAR data), we aim to generate a novel view I^t from a relative viewpoint $P_{r\to t} \in SE(3)$ and RGB camera intrinsics $K \in \mathbb{R}^{3\times3}$. In the latent space, our objective becomes:

$$z^t \sim p(z^t | z^r, D^r, P_{r\to t}, K), \tag{2}$$

where z^t, z^r are the latent representations for I^t, I^r and can be decoded through the VAE's decoder.

3.2. Architecture

Our approach comprises a two-stream architecture, the reference U-Net takes an input view image I^r and produces a feature f^r. Concurrently, the target U-Net takes a noisy latent and generates a novel view image I^t, by integrating the input feature f^r into its internal novel view feature f^t. To provide the diffusion model with the depth-based correspondence, we generate a pair of reference and target point maps $\{X^{r,t}, X^{t,t}\}$ and inject them into the model using ControlNets. An overview of the model's architecture is shown in Fig. 3.

3.3. Pointmap ControlNet.

The advantages of point map have been explored in DUSt3R [45]. Point maps encapsulate the geometry of the scene, the relation between pixels and 3D points, and the relationship between two viewpoints. Such powerful representation can be easily applied to a variety of Multi-View downstream tasks, such as point matching and relative pose

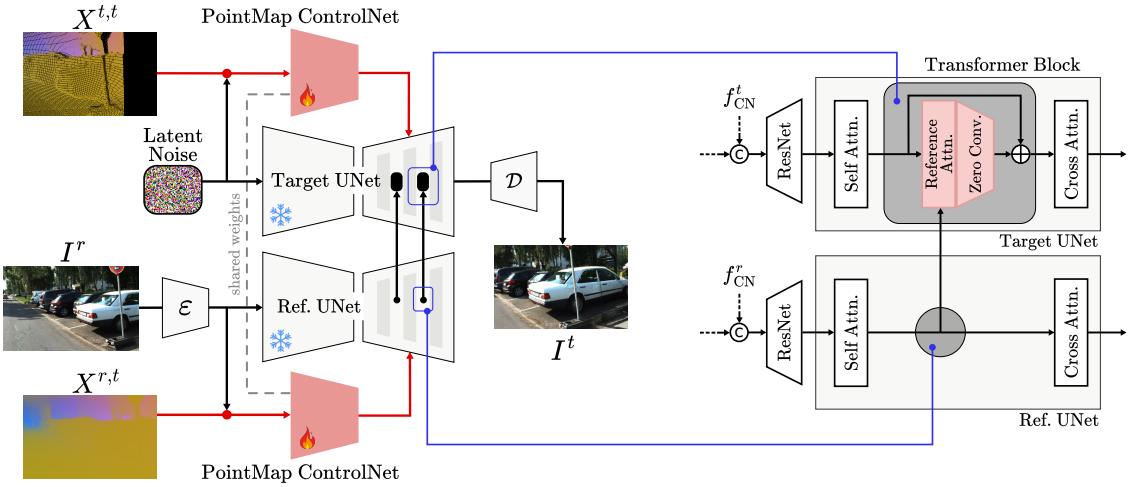


Figure 3. **Method overview.** (left) PointmapDiff is trained in the latent space of a fixed VAE with encoder $\mathcal{E}$ and decoder $\mathcal{D}$. Given a reference RGB image I^r and the corresponding geometry D^r, we obtain a pair of point maps $\{X^{r,t}, X^{t,t}\}$ as inputs. We predict the target image I^t given the geometry signal from the target point map, and information comes from the reference U-Net. Particularly, two Pointmap ControlNets are employed to extract geometric feature correspondences and concatenate © them with the intermediate SD feature maps. We freeze the original SD model and only train the Pointmap ControlNet and the reference attention module. (right) We extract reference features using our reference U-Net. These augmented features are integrated into the target U-Net through a reference-guided cross-view attention mechanism, which is added $\bigoplus$ throughout the target U-Net.

estimation. Since the potential of point maps remains underexplored, this work delves into their advantages within the framework of diffusion models.

We first visit the term *point map*, a point map $X \in \mathbb{R}^{W\times H\times 3}$ describes a mapping between image pixels and 3D scene points. The point map X of the observed scene can be straightforwardly obtained from the camera intrinsic K and the ground-truth depth D as $X_{i,j} = K^{-1}D_{i,j}[i,j,1]^T$, where each pixel represents the projected point coordinate. Here, X is expressed in the camera coordinate frame, but in practice, it can further be denoted as $X^{n,m}$, which is the point map X^n from camera n expressed in camera m's coordinate frame:

$$X^{n,m} = h^{-1}(P_{n\to m}h(X^n)), \tag{3}$$

where $P_{n\to m}$ is the relative pose for images m and n, and $h : (x,y,z) \to (x,y,z,1)$ is the homogeneous mapping. We show that the use of point maps provides robust alignment across views by encoding consistent 3D spatial information; when two pixels across different views share the same point map value, they correspond to the same location in the global coordinate frame. Unlike RGB images, which are sensitive to variations in textures, lighting, and shading, point maps offer explicit and more stable representations, making them particularly beneficial for enforcing consistency in 3D-aware tasks.

Inspired by depth-to-image generation, we utilize ControlNet and inject point maps to enhance the 3D awareness of diffusion features. Specifically, we select pairs of images with known relative camera poses and train the ControlNet to condition on the two point maps $\{X^{r,t}, X^{t,t}\}$. Suppose $\mathcal{F}(\cdot;\Theta)$ is an SD block, with parameters Θ, in particular, the original ControlNet block copies from pre-trained SD's as $\mathcal{F}(\cdot;\Theta')$ and accompanied by two zero convolutions $\mathcal{Z}(\cdot;\Theta_{z1})$, $\mathcal{Z}(\cdot;\Theta_{z2})$. Since the geometric features induced by the point map condition in ControlNet are designed to be aligned with the latent inputs, they are processed through the zero-initialized convolutions and subsequently added to the spatial layers of the U-Net:

$$f^m_{CN} = \underbrace{\mathcal{F}(z^m;\Theta)}_{\text{semantic features}} + \underbrace{\mathcal{Z}\big(\mathcal{F}(z^m + \mathcal{Z}(\gamma(X^{m,t});\Theta_{z1});\Theta');\Theta_{z2}\big)}_{\text{geometric features}}, \tag{4}$$

where f^m_{CN} with $m \in \{r,t\}$ is the set of residual signals, which are augmented with the extracted geometric features to join in the features of the U-Net. The point map is processed using positional encoding [35] function $\gamma(\cdot)$.

These two shared-weight ControlNets help extract the intermediate feature correspondences between the reference and target point maps. Since both point maps are aligned in the same target view coordinate system, the model can naturally follow not only semantic but also geometric correlations between the reference and target views.

3.4. Reference-Guided Cross-View Attention.

Our next step is to learn an attention mechanism between the reference and target features, ensuring consistency between views. We introduce reference attention and inject it

after the self-attention layer in the main target U-Net. This enables the model to capture the corresponding relationships from the reference views. In this module, we change the keys and values corresponding to the output image I^t with those of the reference image I^r. Formally, the output of our reference attention layer is given by:

$$\text{RefAttn}(Q^t, K^r, V^r) = \text{softmax}\left(\frac{Q^t K^{rT}}{\sqrt{d}}\right) \cdot V^r$$
$$\text{with } Q^t = W^Q f^t; K^r = W^K f^r; V^r = W^V f^r, \quad (5)$$

where W^Q, W^K, W^V are learnable projection matrices [41] for the feature inputs f^r, f^t. We further initialize this attention module with the weights from the self-attention module. The output is then passed through a zero convolution layer and added back to the information flow:

$$f = f + \mathcal{Z}\left(\text{RefAttn}(Q^t, K^r, V^r), \Theta_z\right). \quad (6)$$

We further demonstrate the effectiveness of this design in Appendix A.

3.5. Training

Implementation Details. To construct the point maps used in training, we first generate depth maps of reference views using an MDE or depth completion model [45, 61]. Additionally, we incorporate raw LiDAR data to introduce sparse modality by randomly providing point maps derived directly from LiDAR scans for the reference and target views. Using raw LiDAR input, our approach achieves two key benefits: (1) it improves point correspondences for more effective pixel transfer, even when dealing with sparse point distribution, and (2) it encourages the novel view to adhere to the geometric structure imposed by the LiDAR data, leading to more coherent and structure-aware output.

The point maps are then normalized to a fixed range of $[-1, 1]$, followed by a positional encoding. This reduces the model's sensitivity to 3D scale ambiguities. To ensure sufficient correspondence between training pairs, we measure the overlap ratio between two point maps and select only training pairs where the overlap is more than 20%.

Data Augmentation. We apply random cropping to the image to simulate lateral translation. Our method utilizes 3D geometry directly and does not require conditioning on the camera's intrinsic or extrinsic parameters. This strategy helps reduce camera-related ambiguities and allows us to generate training pairs from a single image, making it particularly beneficial for urban driving datasets, where movement is restricted to forward and backward motion.

Training Objective. We leverage the pre-trained SD v1.5 model for both U-Net branches to inherit its generalization ability. We freeze the VAE, the U-Nets, and train only the reference attention modules with the Pointmap ControlNet by minimizing the following cost function:

$$\mathcal{L}(\theta) = \mathbb{E}_{\epsilon,\tau}\left[\|\epsilon - \epsilon_\theta(z_\tau^t, z^r, X^{r,t}, X^{t,t}, P_{r\to t}, K, \tau)\|_2^2\right] \quad (7)$$

on a dataset containing pairs of source view image I^r, target view image I^t that are encoded into z^r, z^t respectively, their camera information $\{P_{r\to t}, K\}$, and point maps $\{X^{r,t}, X^{t,t}\}$. We adopt DDIM sampler [33] and add noise at τ to the target latent z_τ^t while keeping the reference latent z^r clean. We use AdamW optimizer [17] with a learning rate of 1e-4, applying cosine scheduler and train the model for 500k iterations with a batch size of 4 at the resolution of 512×320. Other training parameters remain set to their default values. During inference, we fix the sampling steps at 50.

4. Experiments

We prepare two experiments to evaluate our model. First, we demonstrate how PointmapDiff can be used to enhance the rendering quality of 3DGS [13] at extrapolated viewpoints. Second, we assess the model's performance without 3DGS, showcasing its ability to generate views from a single input. We train and evaluate our model on the KITTI-360 [15] and Waymo [34] datasets.

4.1. Extrapolation in Street View Reconstruction

We fine-tune 3DGS on synthesized views to enhance the rendering quality on extrapolated views. We construct evaluation sets that (1) look left and right by rotating the camera by $\pm 45°$ around the camera's y-axis, (2) shift the camera lateral to the heading direction by $\pm\{2, 4\}m$, and (3) look downward by rotating the camera by $10°$ around the camera's x-axis while flying upward by $1m$.

3D Gaussian Splatting. We first prepare a pre-trained 3DGS on available ground truth images for 20000 iterations. We refer to the original 3DGS paper [13] for the fundamental training pipeline. Next, rendered images from augmented viewpoints are encoded and perturbed into noisy latents based on a noise scale $s = \frac{\tau}{T}$. PointmapDiff is used to refine these noisy novel-view images, which are then added to the training data. We choose the reference image with the highest overlap and the closest training LiDAR scan as the condition. We continue fine-tuning the same 3DGS model for another 20000 iterations, in which for every 200 steps, we repeatedly update the augmented set with a linearly reduced s. The scene representation is optimized using two different loss functions:

$$\mathcal{L}_{train} = \lambda_{rgb}\mathcal{L}_{rgb} + \lambda_{ssim}\mathcal{L}_{ssim} + \lambda_d\mathcal{L}_d, \quad (8)$$
$$\mathcal{L}_{aug} = \lambda_{aug}\mathcal{L}_{rgb} + \lambda_{lpips}\mathcal{L}_{lpips} + \lambda_d\mathcal{L}_d, \quad (9)$$

for ground truth training data $\mathcal{L}_{train}$ and augmented data $\mathcal{L}_{aug}$, where $\mathcal{L}_{rgb}$, $\mathcal{L}_{ssim}$ denote the L1 and SSIM losses,



Figure 4. Qualitative comparison on KITTI-360 [15]. We demonstrate three scenarios: rotating, shifting, and flying upward movement. The test view represents the conventional camera sampled from forward-facing trajectories. We also include training images that provide the best available coverage as a reference.

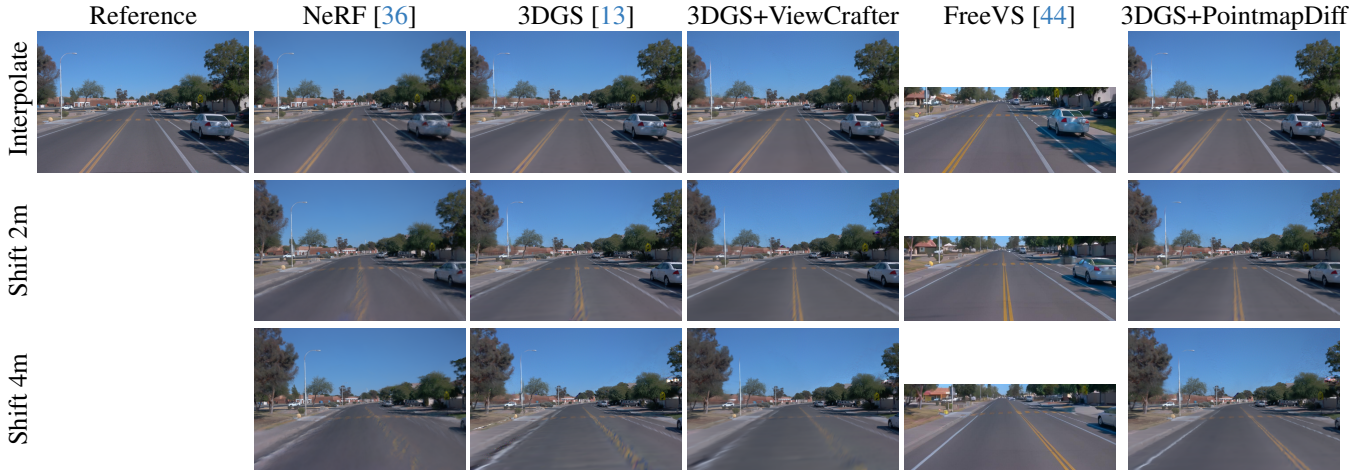


Figure 5. Qualitative comparison on Waymo [34] with different shifting distances.

respectively. We treat augmented views differently by lowering the weight of L1 loss and incorporating LPIPS loss $\mathcal{L}_{lpips}$ between 3DGS-rendered and diffusion-generated images to prioritize high-level semantic similarity over strict photometric consistency [7]. Furthermore, we utilize depth loss $\mathcal{L}_d = \|\hat{D} - D_{lidar}\|_1$ across all views, as the extrapolated views are designed to be geometrically aligned with the LiDAR data. More details can be found in the supplementary material.

Baselines. We compare our method with existing state-of-the-art urban driving scene reconstruction and extrapolation methods, including SGD [58], VEGS [12], ViewCrafter [57] and FreeVS [44] as well as original 3DGS [13] and Nerfacto [36]. For a fair comparison, we allow 3DGS and incorporate accumulated LiDAR data for Gaussian initialization and apply depth supervision to both 3DGS and Nerfacto. Furthermore, we redefine the set of augmented samples for SGD and VEGS to align with our mentioned evaluation criteria.

Metrics. We select every 8^{th} frame as conventional test frame for interpolation. We adopt PSNR, SSIM [46], and LPIPS [60] and evaluate at resolution 1408×376 for KITTI-360 and 960×640 for Waymo. On the other hand, we report FID [8] and KID ($\times100$) [3] under extrapolation settings since no ground truth image is available. In these cases, KITTI's images are cropped to 600×376, ensuring there is

not a lot of unobserved space that could disturb the results, following Hwang *et al*. [12].

	Interpolation			Extrapolation	
	PSNR↑	SSIM↑	LPIPS↓	FID↓	KID↓
Nerfacto [36]	22.02	0.731	0.270	101.60	5.858
3DGS [13]	23.44	0.792	0.122	85.08	4.926
SGD [58]	23.23	0.779	0.162	82.77	4.207
VEGS [12]	22.77	0.790	0.143	80.79	4.273
3DGS+PointmapDiff	23.39	0.790	0.144	78.07	3.799

Table 1. Quantitative results for interpolated and extrapolated view synthesis on KITTI-360 [15]. Label with best, second, and third .

	Interpolation			Extrapolation	
	PSNR↑	SSIM↑	LPIPS↓	FID↓	KID↓
Nerfacto [36]	29.75	0.884	0.189	102.79	5.650
3DGS [13]	30.18	0.914	0.076	62.50	2.401
3DGS+ViewCrafter [57]	29.83	0.910	0.078	54.16	2.035
FreeVS [44]	23.37	0.743	0.180	51.95	1.838
3DGS+PointmapDiff	30.45	0.913	0.079	48.22	1.035

Table 2. Quantitative results for interpolated and extrapolated view synthesis on Waymo [34].

As illustrated in Fig. 4 and Fig. 5, all methods demonstrate improvements over the original 3DGS and NeRF on extrapolated viewpoints. However, while SGD reduces the artifacts, it does not completely resolve the issue, as some blur and traces of artifacts still appear in the generated views. VEGS performs well on planar regions such as roads and buildings, thanks to normal supervision, but struggles with far-away background. ViewCrafter works effectively with small viewpoint shifts but underperforms on larger displacements. FreeVS slightly suffers from color change and is limited to areas where LiDAR data is available, which affects its performance in distant areas and sky regions. On the other hand, our method portrays clear improvement across all scenarios. We report quantitative comparison in Tab. 1 and Tab. 2. Combining 3DGS with PointmapDiff outperforms the baselines in inception distances while retaining the most rendering quality of conventional test cameras.

4.2. Single-image NVS on Street View

Setup and Baselines. In this experiment, we use a single source image as input to predict target images corresponding to 4-6 previous or following frames in the trajectory. Our baselines include warping and inpainting method using SD Inpainting [27], GenWarp [30], and BTS [47] as one-shot implicit NVS. To ensure scene-specific visual fidelity, we fine-tune SD using LoRA [10] on the same dataset. Since the baselines are not designed to handle LiDAR inputs directly, we condition with completed depth maps [61] instead. We use FID [8] and KID ($\times 100$) [3] to estimate the realism of the generated image distribution; depth metrics (absolute relative error, root mean square error, and threshold accuracy $\delta < 1.25$) between predicted depth [11] of the generated images and LiDAR depth to measure the geometric consistency. We also include indoor experiments on RealEstate10K [62] and ScanNet++ [53] with more competitors in the supplementary material.

	FID↓	KID↓	AbsRel↓	RMSE↓	δ_1 ↑
Inpainting [27]	34.61	0.789	0.241	6.496	61.99
GenWarp [30]	45.41	2.080	0.270	6.563	56.05
BTS [47]	74.66	6.120	0.351	7.131	44.33
PointmapDiff (ours)	28.31	0.392	0.185	5.030	73.36
Lower bound	11.61	0.028	0.118	3.831	87.51

Table 3. Quantitative results for singe-image NVS on KITTI-360 [15]. We include the inception distances between the source and the target view and the predicted depth of the ground truth target view as the lower bound.

Fig. 6 and Tab. 3 show that PointmapDiff demonstrates the ability to follow the scene geometry to effectively generate plausible high-quality views, whereas GenWarp and BTS suffer heavily from stretching artifacts. We also outperform in most metrics.

4.3. Ablation Study

To study the design of PointmapDiff, we conduct single-view NVS experiment on all the model variants. Tab. 4 and Fig. 7 demonstrate the results of our study.

Ablation	FID↓	KID↓	AbsRel↓	RMSE↓	δ_1 ↑
w/o ControlNet	39.50	0.936	0.422	10.937	39.70
w/o Attention	64.57	3.626	0.272	6.899	56.61
w/o Pointmap P.E.	24.80	0.276	0.184	5.009	72.48
Full model	21.26	0.268	0.181	4.880	74.46

Table 4. Quantitative ablation of individual components.

Pointmap ControlNets. When excluding the Pointmap ControlNets, the model loses access to the correct correspondences derived from the reference image. This omission impairs its ability to maintain spatial consistency, resulting in generated views that respect the reference appearance but deviate significantly in geometry (Fig. 7d).

Reference-Guided Cross-View Attention. Without reference attention, the model operates similarly to a standard geometry-controlled SD model. Even when we employ LLaVA [16] to input more detailed scene descriptions, the model struggles to respect the contents. However, this version provides valuable insights; specifically, it demonstrates that point maps are an effective conditioning source. They successfully encode the scene's geometry, helping the

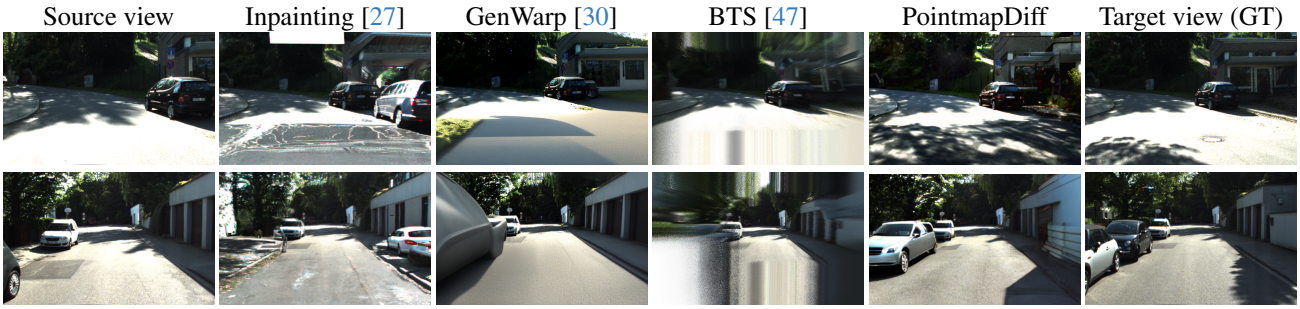


Figure 6. Qualitative comparison for single-image NVS on KITTI-360 [15].



(a) Source view (b) Target view (c) Full model



(d) w/o Pointmap CN (e) w/o Reference Attn. (f) w/o Pointmap P.E.

Figure 7. The full model effectively captures high-detail scene continuity, closely aligning with the target image, while removing components leads to a loss in both geometric structure or appearance fidelity.

model recover the scene's structure reasonably, only without precise adherence to the reference image. In Fig. 7e, while the model can place the cars in the correct position, it fails to transfer the appearance from the source image accurately.

Pointmap Positional Encoding (P.E.). Fig. 7f shows that directly passing point map coordinates into the network results in reduced image detail (*e.g.*, texture on the road and the shadow regions) and lower metrics, whereas preprocessing the input with positional embedding (Fig. 7c) enables the model to represent higher frequency details.

4.4. Discussion

In this section, we explore additional capabilities granted by the architecture design of PointmapDiff, along with its current limitations.

LiDAR-algined Generation. PointmapDiff demonstrates a remarkable ability to generate images that align closely with LiDAR-derived conditions, preserving spatial structure and depth cues inherent in point cloud data. This is critical for autonomous driving applications, where hallucinated regions must remain consistent with real-world geometry and context, rather than being entirely random. Such fidelity ensures that generated content supports safe and reliable perception in driving environments. We refer to the



Source view Target view Translation Duplication

Figure 8. **Scene editing results.** We perform instance-level editing by manipulating point map values.

Appendix C.3 for more illustrations.

Scene Editing. Our model allows image editing by manipulating the point map, which enables repositioning or duplicating objects within a scene without changing their visual appearances. We first isolate the points belonging to the objects of interest using 3D bounding boxes or instance labels. Then, spatial transformations are applied to these points while keeping their initial values in the point map. This helps the model to establish correspondences based on these transformations. Following this idea, we showcase in Fig. 8 two scenarios where we shift and duplicate a set of points that belong to a car. As a result, we can perform both novel view synthesis and spatial editing at the same time. This provides a promising direction for future explorations in scene manipulation through point map-based editing.

Limitations. While PointmapDiff improves extrapolated views, the model can still introduce inconsistencies that cause blurriness, thus lower reconstruction and interpolation qualities. Additionally, the current design is optimized for static scenes and may struggle with dynamic environments. Future work will focus on extending PointmapDiff to better handle dynamic scenarios, incorporating temporal consistency and object tracking to improve generation quality in more complex settings.

5. Conclusion

We present a novel approach for view synthesis by integrating point map-conditioned diffusion, significantly improving scene reconstruction in extrapolating use cases. Our method enhances spatial consistency and enables high-quality novel view generation. Detailed evaluations confirm its robustness and effectiveness in urban driving data and open new editable view synthesis applications.

References

[1] Yuval Alaluf, Daniel Garibi, Or Patashnik, Hadar Averbuch-Elor, and Daniel Cohen-Or. Cross-image attention for zero-shot appearance transfer. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–12, 2024. 12

[2] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 2, 16

[3] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 6, 7, 14

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 16

[5] Shengqu Cai, Eric Ryan Chan, Songyou Peng, Mohamad Shahbazi, Anton Obukhov, Luc Van Gool, and Gordon Wetzstein. Diffdreamer: Towards consistent unsupervised single-view scene extrapolation with conditional diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2139–2150, 2023. 16

[6] Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *arXiv preprint arXiv:2311.13384*, 2023. 16

[7] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024. 2, 6, 12

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6, 7, 14

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[10] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 3, 7

[11] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 7, 13

[12] Sungwon Hwang, Min-Jung Kim, Taewoong Kang, Jayeon Kang, and Jaegul Choo. Vegs: View extrapolation of urban scenes in 3d gaussian splatting using learned priors. In *European Conference on Computer Vision*, pages 1–18. Springer, 2024. 3, 6, 7, 13

[13] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1, 2, 5, 6, 7, 13, 14

[14] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[15] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022. 5, 6, 7, 8, 13, 17

[16] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. 7

[17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[18] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2

[19] Norman Müller, Katja Schwarz, Barbara Rössle, Lorenzo Porzi, Samuel Rota Bulò, Matthias Nießner, and Peter Kontschieder. Multidiff: Consistent novel view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10258–10268, 2024. 2

[20] Thang-Anh-Quan Nguyen, Luis Roldão, Nathan Piasco, Moussab Bennehar, and Dzmitry Tsishkou. Rodus: Robust decomposition of static and dynamic elements in urban scenes. In *European Conference on Computer Vision*, pages 112–130. Springer, 2024. 1

[21] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2856–2865, 2021. 1

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3

[23] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020. 2, 16

[24] Xuanchi Ren and Xiaolong Wang. Look outside the room: Synthesizing a consistent long-term 3d scene video from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3563–3573, 2022. 2, 14

[25] Chris Rockwell, David F Fouhey, and Justin Johnson. Pixelsynth: Generating a 3d-consistent experience from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14104–14113, 2021. 16

[26] Robin Rombach, Patrick Esser, and Björn Ommer. Geometry-free view synthesis: Transformers and no 3d priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14356–14366, 2021. 13, 15, 16

[27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3, 7, 8, 13, 14, 15, 16

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 3

[29] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. 15, 16

[30] Junyoung Seo, Kazumi Fukuda, Takashi Shibuya, Takuya Narihira, Naoki Murata, Shoukang Hu, Chieh-Hsin Lai, Seungryong Kim, and Yuki Mitsufuji. Genwarp: Single image to novel views with semantic-preserving generative warping. *Advances in Neural Information Processing Systems*, 37:80220–80243, 2025. 2, 7, 8, 12, 13, 14, 15, 16

[31] Jaidev Shriram, Alex Trevithick, Lingjie Liu, and Ravi Ramamoorthi. Realmdreamer: Text-driven 3d scene generation with inpainting and depth diffusion. *arXiv preprint arXiv:2404.07199*, 2024. 16

[32] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *Advances in Neural Information Processing Systems*, 34: 19313–19325, 2021. 2

[33] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 5

[34] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 5, 6, 7, 13, 14

[35] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020. 4

[36] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–12, 2023. 6, 7, 13, 14

[37] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *Advances in Neural Information Processing Systems*, 36:1363–1389, 2023. 12

[38] Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. *arXiv preprint arXiv:2307.01097*, 2023. 2

[39] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Wang Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. Advances in neural rendering. In *Computer Graphics Forum*, pages 703–735. Wiley Online Library, 2022. 2

[40] Hung-Yu Tseng, Qinbo Li, Changil Kim, Suhib Alsisan, Jia-Bin Huang, and Johannes Kopf. Consistent view synthesis with pose-guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16773–16783, 2023. 2

[41] Ashish Vaswani. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 5

[42] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12619–12629, 2023. 3

[43] Haiping Wang, Yuan Liu, Bing Wang, Yujing Sun, Zhen Dong, Wenping Wang, and Bisheng Yang. Freereg: Image-to-point cloud registration leveraging pretrained diffusion models and monocular depth estimators. *arXiv preprint arXiv:2310.03420*, 2023. 3

[44] Qitai Wang, Lue Fan, Yuqi Wang, Yuntao Chen, and Zhaoxiang Zhang. Freevs: Generative view synthesis on free driving trajectory. *arXiv preprint arXiv:2410.18079*, 2024. 3, 6, 7, 14

[45] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 3, 5, 14

[46] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[47] Felix Wimbauer, Nan Yang, Christian Rupprecht, and Daniel Cremers. Behind the scenes: Density fields for single view reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9076–9086, 2023. 7, 8

[48] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21551–21561, 2024. 2

[49] Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao

Chen, Runyi Yang, et al. Mars: An instance-aware, modular and realistic simulator for autonomous driving. In *CAAI International Conference on Artificial Intelligence*, pages 3–15. Springer, 2023. 1

[50] Chenfeng Xu, Huan Ling, Sanja Fidler, and Or Litany. 3difftection: 3d object detection with geometry-aware diffusion features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10617–10627, 2024. 3

[51] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *European Conference on Computer Vision*, pages 156–173. Springer, 2024. 1

[52] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, et al. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023. 1

[53] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023. 7, 13, 15, 16

[54] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4578–4587, 2021. 2

[55] Jason J Yu, Fereshteh Forghani, Konstantinos G Derpanis, and Marcus A Brubaker. Long-term photometric consistent novel view synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7094–7104, 2023. 2, 13

[56] Jason J Yu, Tristan Aumentado-Armstrong, Fereshteh Forghani, Konstantinos G Derpanis, and Marcus A Brubaker. Polyoculus: Simultaneous multi-view image-based novel view synthesis. *arXiv preprint arXiv:2402.17986*, 2024. 2

[57] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024. 2, 6, 7

[58] Zhongrui Yu, Haoran Wang, Jinze Yang, Hanzhang Wang, Jiale Cao, Zhong Ji, and Mingming Sun. Sgd: Street view synthesis with gaussian splatting and diffusion prior. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3812–3822. IEEE, 2025. 3, 6, 7, 13

[59] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 3

[60] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6, 14

[61] Youmin Zhang, Xianda Guo, Matteo Poggi, Zheng Zhu, Guan Huang, and Stefano Mattoccia. Completionformer: Depth completion with convolutions and vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18527–18536, 2023. 5, 7

[62] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 7, 13, 15, 16

[63] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21634–21643, 2024. 1

Pointmap-Conditioned Diffusion for Consistent Novel View Synthesis

Supplementary Material

A. Implementation Details

A.1. Design Motivation

We further explain the motivation for using point maps as conditioning signals. Given reference r and target t viewpoints, establishing correspondences $\mathcal{M}^{r,t}$ between pixels of two images can be trivially achieved by nearest neighbor (NN) search in the 3D point map space:

$$\mathcal{M}^{r,t} = \left\{(a,b) \mid a = \text{NN}^{t,r}(b) \text{ and } b = \text{NN}^{r,t}(a)\right\},$$
$$\text{with } \text{NN}^{m,n}(a) = \underset{b\in\{0,\dots,WH\}}{\arg\min} \left\|X_b^{n,n} - X_a^{m,n}\right\|. \quad (10)$$

Here, $\text{NN}^{m,n}$ computes the nearest neighbor b of pixel a between views m and n. While this explicit correspondence is computationally expensive and only operates on pixel space, it motivates our approach of leveraging implicit attention mechanisms.

We consider a simple positional encoding example of point maps, $\gamma(X)$, which maps the normalized input points to higher-dimensional Fourier features using a set of sine and cosine functions:

$$\gamma(\mathbf{x}) = [a_1\cos(2\pi F_1\mathbf{x}), a_1\sin(2\pi F_1\mathbf{x}), \dots, a_N\cos(2\pi F_N\mathbf{x}), a_N\sin(2\pi F_N\mathbf{x})]^T, \quad (11)$$

where F_j are the Fourier basis frequencies and a_j are their corresponding coefficients. Using this encoding, the spatial correlation between two point maps can be measured via a kernel function as:

$$\gamma(\mathbf{x}_1)\gamma(\mathbf{x}_2)^T = \sum_{j=1}^{N} a_j^2\cos\left(2\pi F_j(\mathbf{x}_1 - \mathbf{x}_2)\right). \quad (12)$$

To adapt this to the nearest neighbor computation, we redefine $\text{NN}^{m,n}$ using the encoded point maps as follows:

$$\text{NN}^{m,n}(a) = \underset{b\in\{0,\dots,WH\}}{\arg\max} \left(\gamma\left(X_b^{n,n}\right)\gamma\left(X_a^{m,n}\right)^T\right), \quad (13)$$

by replacing $t \leftarrow n$ and $r \leftarrow m$, and applying this for all $a \in \{0,\dots,WH\}$, interestingly, this operation resembles the reference attention mechanism introduced in the main paper. Specifically, the attention matrix: $A = \text{softmax}\left(\frac{Q^t K^{r\,T}}{\sqrt{d}}\right)$ serves a similar purpose by learning implicit correspondences between the query (Q^t) and key (K^r) representations extracted from Pointmap ControlNet's layers of the target and reference views. Thus, the point map conditioning acts as an intermediate signal to naturally establish correspondences within the attention layers, bridging the gap between explicit point matching and feature-based reasoning with the ability to dynamically attend to relevant regions. Hence, we verify the roles of the keys and queries in Fig. 9; they determine the regions in the source views that can be used for generation.



Figure 9. Given a query point in the upper-left generated view and reference views, we extract PointmapDiff's intermediate layer activations through the keys and queries of self-attention and reference attention layers at a certain time step $\tau = 0.2T$ during the denoising process and use them to visualize the attention maps [1, 37]. As a result, the method can learn and produce correct correspondences.

A.2. Training

Our method employs a pre-trained SD v1.5 as the backbone, thanks to its robust generative capabilities. Since SD v1.5 is also a text-to-image model, we incorporate simple text prompts, such as *"a photo of a driving scene"* or *"a photo of a room"*, to provide high-level semantic guidance.

Unlike other methods [7, 30], we do not modify the latent input, allowing us to retain the U-Net backbone and instead adapt to the task by training the additional ControlNet. For the positional encoding, we use a frequency range from 2^0 to 2^3, resulting in an input channel dimension of 24 for the ControlNet model. As the training progresses, we observe sudden convergence of ControlNet after approximately 10K iterations, portrayed in Fig. 10. The model continues to refine fine-grained details beyond this point, yielding steady improvements in PSNR and SSIM.

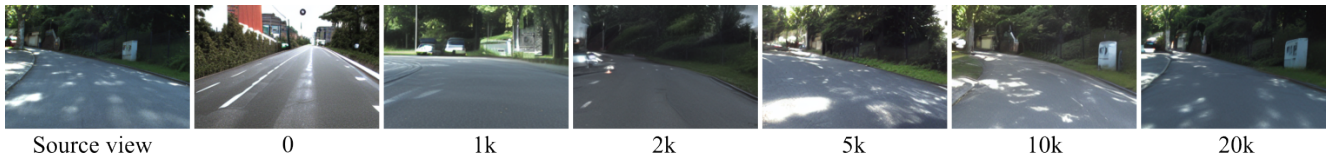


Figure 10. Validation sample observed in several training iterations.

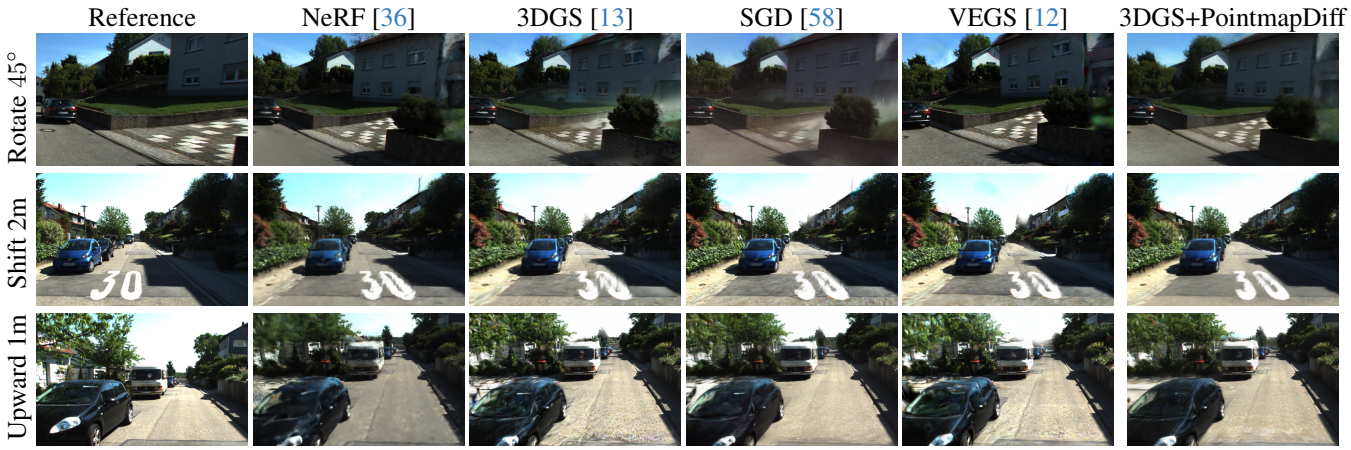


Figure 11. Additional qualitative comparison on KITTI-360 [15].

A.3. Optimizing 3DGS

We randomly select 10 static sub-sequences per dataset, each with 100-150 consecutive frames for evaluation. For KITTI-360, we train on two perspective images with full resolution 1408×376, and for Waymo, we only use the front camera downsampled to 960×640. We initialize the 3D Gaussian models with the accumulated LiDAR point cloud without using Structure-from-Motion (SfM) point clouds.

The loss weights λ_{rgb}, λ_{ssim}, λ_{aug}, λ_{lpips}, and λ_d are set to 0.8, 0.2, 0.5, 0.1, and 0.01, respectively. Additionally, we progressively reduce the noise scale s from 0.6 to 0.2 throughout training to ensure harmonization between generation and reconstruction.

B. Additional Results

B.1. Extrapolation in Street View Reconstruction

Baseline Implementations. We adapt the code in the official repository of VEGS, ViewCrafter, and FreeVS. We re-implement SGD since there is no public code base available. For ViewCrafter, we use ground truth images and rendered depth from 3DGS to achieve warped conditions, since predictions from MDE are extremely noisy and not aligned well with the shifted distance. Secondly, as ViewCrafter is a video diffusion model, and requires the first frame to be "clean" (*i.e.*, from ground truth trajectory), we design shifting samples, gradually from the original to the novel trajectory to extract the most details from this initial frame. Since the sequences contain more frames than a video diffusion model can handle at once, the process is divided into smaller chunks and repeated across the entire sequence. The distillation process for ViewCrafter remains mostly the same as with PointmapDiff. We show additional qualitative comparisons in Fig. 11 and Fig. 12.

B.2. Single-image NVS on Street View

We provide results for single-image NVS task on Waymo [34] in Fig. 13. We utilize Metric3D [11] to estimate depth, as there is no reliable depth completion model available for Waymo.

B.3. Single-image NVS on Indoor Data

Baselines. These include GeoGPT [26], Photoconsistent-NVS [55], the warping and inpainting method using the SD Inpainting [27], and GenWarp [30]. To ensure fair comparisons, we train our model only on RealEstate10K [62], aligning with the training data used by our baselines, and further evaluate on ScanNet++ [53] to assess performance on out-of-distribution scenarios. We use the officially provided checkpoint of all methods. For GeoGPT, we choose the `re_impl_depth` checkpoint as it requires reference depth maps and produces better results compared to the version that does not use depth information. Moreover, for SD-Inpainting, we apply interpolation on the warped images and dilate the inpaint mask using a 9×9 kernel to reduce artifacts since the model performs inpainting on latent

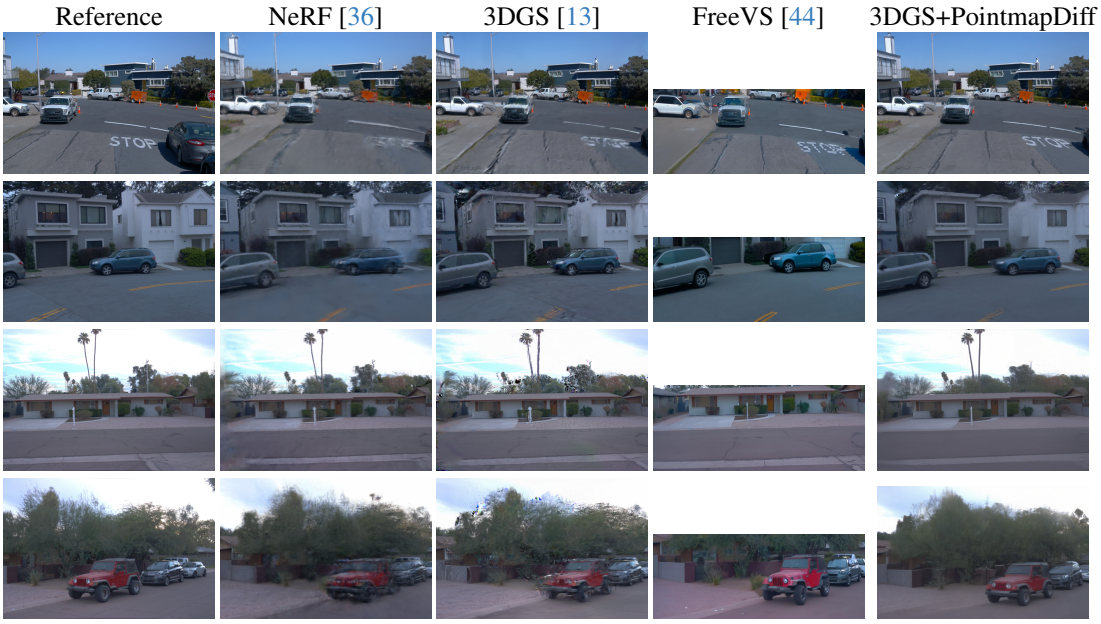


Figure 12. Addtional qualitative comparison on Waymo [34].



Figure 13. Qualitative comparison for single-image NVS on Waymo [34].

space (*f8d4*). In contrast, only PhotoNVS does not require depth as an input.

Setup. We utilize DUSt3R [45] to generate point maps for training and as a depth estimator (by taking the z-values of the point map in local coordinate) for inference of all baselines. Similar to [24, 30], we consider dividing into short-term and long-term view synthesis. Specifically, we randomly select 1k sequences from the test set with more than 200 frames and evaluate the 50^{th} generated frame as short-term and the 100^{th} generated frame as long-term view synthesis on RealEstate10K. Due to the faster camera movement in ScanNet++, we focus solely on short-term synthesis.

Metrics. For short-term, we use pairwise reconstruction metrics PSNR and LPIPS [60] to measure the difference between the generated and ground-truth images. Note that these metrics become less relevant in regions that are unseen. For long-term, we value generated image quality, using the FID [8] and KID ($\times 100$) [3] to estimate the realism of the generated image distribution. Finally, all outputs are resized and cropped to 256×256 for evaluation.

Tab. 5 demonstrates that while GeoGPT achieves good FID and KID, indicating realistic generation quality, it struggles with misalignment issues from the input view, leading to worse PSNR and LPIPS scores. In contrast, the inpainting method excels in PSNR, benefiting from explicit warping. However, it often suffers from artifacts due to the imperfect depth, resulting in lower FID and KID.

For the out-of-distribution experiment, as shown in Tab. 6, GeoGPT and Photoconsistent-NVS struggle to generalize to out-of-domain scenarios, resulting in poor performance metrics and a noticeable drop in generation quality. On the other hand, our method achieves stable and consistent results across both in-domain and out-of-domain

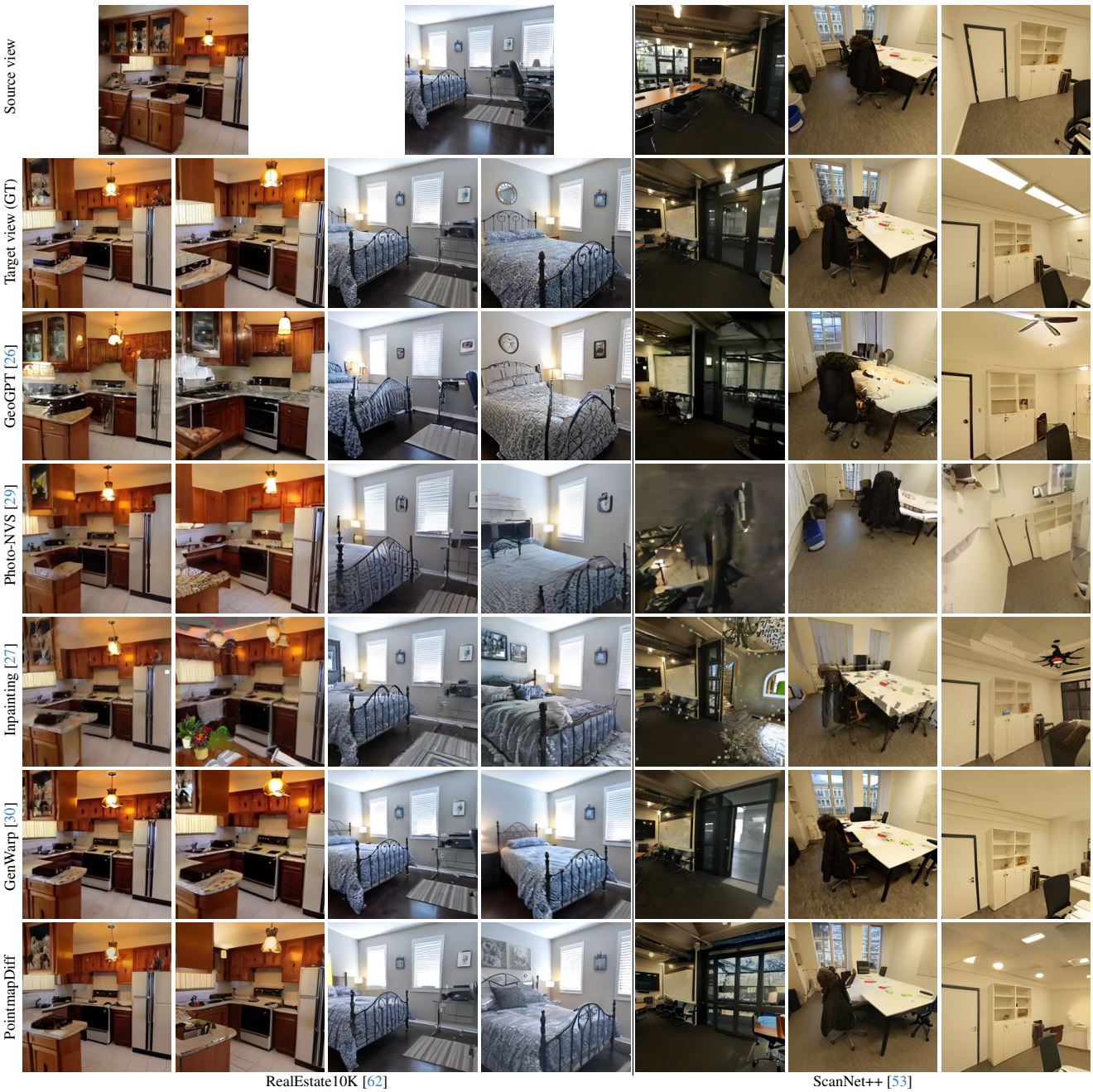


Figure 14. Additional NVS comparison on RealEstate10K [62] and ScanNet++ [53].

datasets, indicating improved adaptability and maintaining high-quality view synthesis under diverse conditions while mitigating overfitting.

Fig. 14 shows qualitative comparisons on RealEstate10K and ScanNet++. The inpainting method performs well in regions where there is a clear overlap between the input and the novel views. However, in areas with sparse warped pixels, it produces inconsistent novel views, failing to take into account the information from the surrounding input pixels, which impacts the overall coherence. Our method consistently synthesizes realistic and stable novel views across both small and large viewpoint changes, compatible with the quality of GenWarp despite training on less data.

	Short-term		Long-term	
	PSNR↑	LPIPS↓	FID↓	KID↓
GeoGPT [26]	14.97	0.356	28.42	0.158
Photo-NVS [29]	15.74	0.309	30.96	0.305
Inpainting [27]	16.29	0.300	47.63	1.546
PointmapDiff	16.04	0.272	32.34	0.446
GenWarp [30]	15.87	0.237	29.65	0.446

Table 5. Quantitative results on RealEstate10K [62].

	Short-term			
	PSNR↑	LPIPS↓	FID↓	KID↓
GeoGPT [26]	14.50	0.328	62.70	2.256
Photo-NVS [29]	11.72	0.525	90.05	4.143
Inpainting [27]	15.09	0.312	56.08	1.647
GenWarp* [30]	15.95	0.248	29.63	0.336
PointmapDiff	15.19	0.303	38.72	0.560

Table 6. Quantitative comparisons with SoTA methods on Scannet++ [53]. GenWarp achieves slightly better results because it is trained on datasets beyond RealEstate10K.

C. Additional Analysis

C.1. Multi-View Conditioning

Our method can be easily extended to condition on a set of multiple reference images, $\{I^{r_1}, \ldots, I^{r_k}\}$. This is achieved by concatenating the keys and values from all the reference images, as all point maps share the same coordinate system (i.e., the target coordinate). This allows the model to naturally integrate information from multiple reference views and inherently decide which views it should rely more on during generation, enhancing the quality and consistency of the output. Formally, the key and value with multiple images guidance are obtained with the following expressions:

$$K^r = W^K[f^{r_1}, \ldots, f^{r_k}]; V^r = W^V[f^{r_1}, \ldots, f^{r_k}]. \quad (14)$$

While our model has been trained using only one reference view, it is worth emphasizing that it can benefit from multiple reference view conditioning without further fine-tuning or modification. This approach helps reconstruct by leveraging details visible in alternate views, resulting in a more coherent and complete scene, as shown in Fig. 15.

C.2. Robust to Noisy Depth

Additionally, when leveraging off-the-shelf MDE models [2, 23], the generated depth maps D^r used for wrapping and establishing point correspondences can be noisy. However, our reference attention mechanism additionally injects both semantic and geometric multi-resolution information from the reference image as a guiding signal. This enables the model to be naturally more robust to noisy or incomplete depth within the generative prior compared to the explicit warping [5, 6, 25, 31] approaches. We show in Fig. 16 a scenario where using monocular depth can lead to ill-warping artifacts and Fig. 17 where sparsity of LiDAR points makes inpainting infeasible. As said, PointmapDiff demonstrates a strong ability to fill in missing regions and correct inaccurate geometry, highlighting its capacity to understand scene structure without overfitting to misaligned inputs.

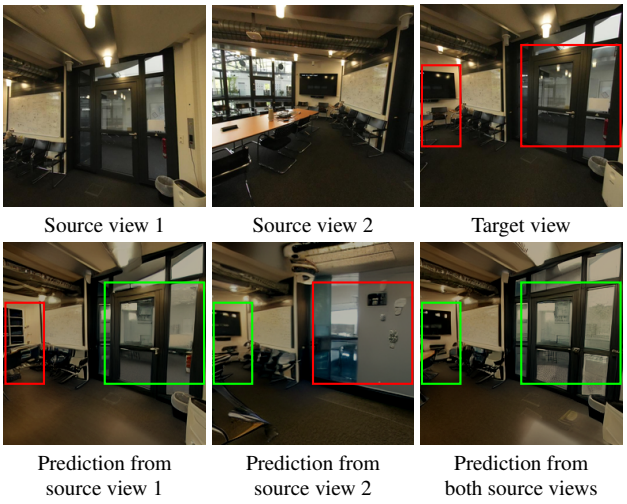


Figure 15. We demonstrate the results when generating viewpoints between two source views, effectively covering occluded regions by combining complementary information from both views. We use red to denote hallucinated regions and green to indicate aligned regions compared to the target view.

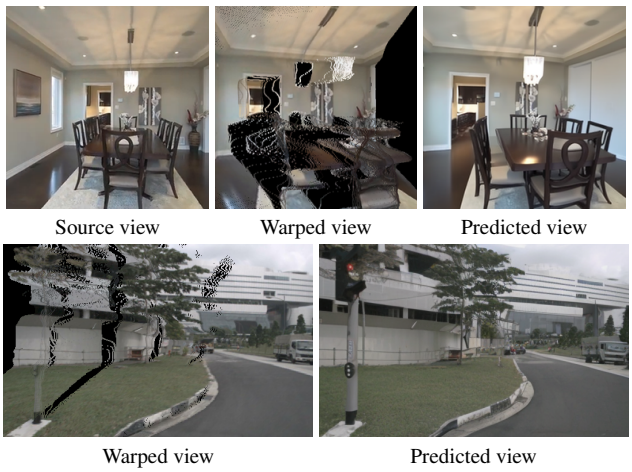
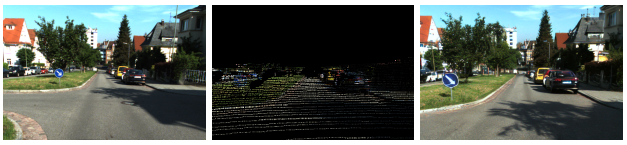


Figure 16. Robustness to noisy depth on RealEstate10K [62] and nuScenes [4].



Source view Warped view Predicted view

Figure 17. Robustness to sparse depth on KITTI-360 [15].



Source views Predicted views

Figure 18. We overlap projected LiDAR points onto the images on KITTI-360 [15], showing that our generated views are aligned with the geometry given by the LiDAR.

C.3. LiDAR-aligned Generation

Fig. 18 shows that by integrating LiDAR data from regions not visible in the reference views, we can generate images that accurately adhere to the underlying LiDAR measurements, ensuring high-fidelity scene reconstruction with enhanced geometric consistency.

C.4. Limitations and Future Work

In this section, we discuss the primary limitations of our work and propose some preliminary mitigation strategies for future research. The diffusion model is trained to remove noise, but stochasticity persists in the final prediction. Moreover, lossy compression of VAE can remove contents in the prediction, particularly in small details. When using these images to train 3DGS, this can lead to blurry results, even in regions that are well-observed in the training ground truths, leading to lower PSNR and SSIM during interpolation. An interesting focus for future work would be to study the uncertainty in both the 3DGS and diffusion models. This involves updating only the regions where 3DGS is uncertain, while the diffusion model is confident, and vice versa. Additionally, to adapt to dynamic scenes, it is necessary to introduce a temporal dimension to the diffusion model. This approach, commonly used in video diffusion, could complement object movement where static point maps cannot provide correct correspondences.