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.

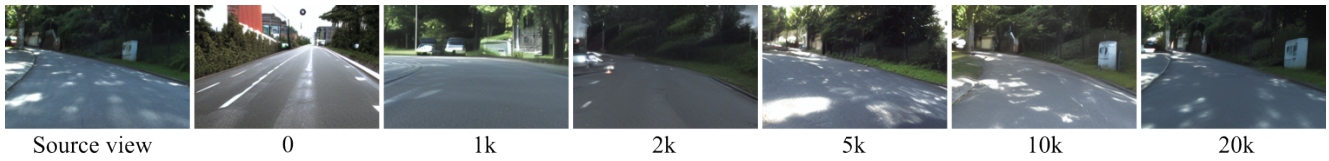


Figure 10. Validation sample observed in several training iterations.

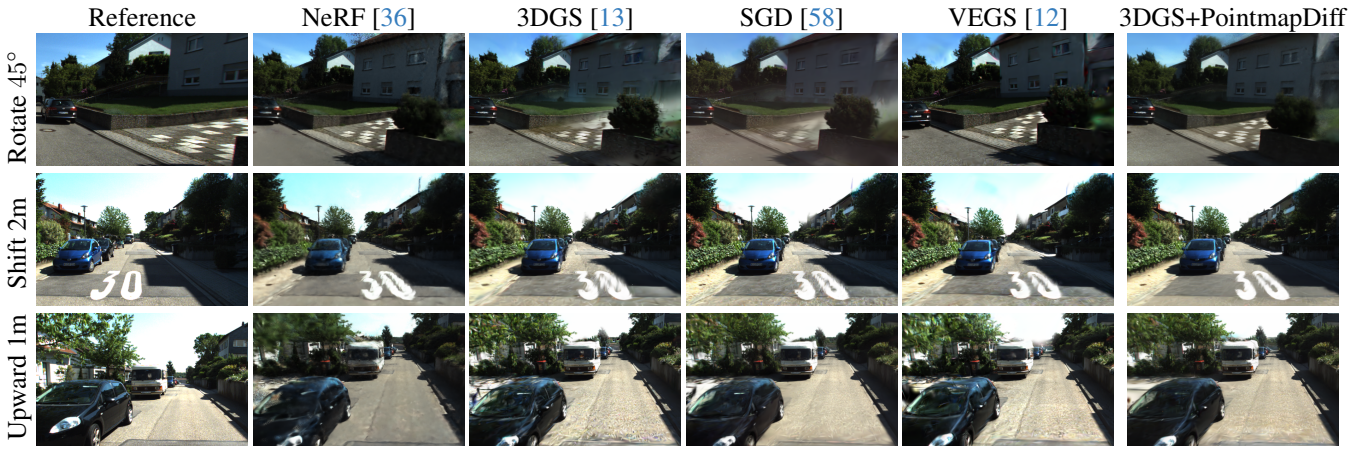


Figure 11. Additional qualitative comparison on KITTI-360 [15].

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

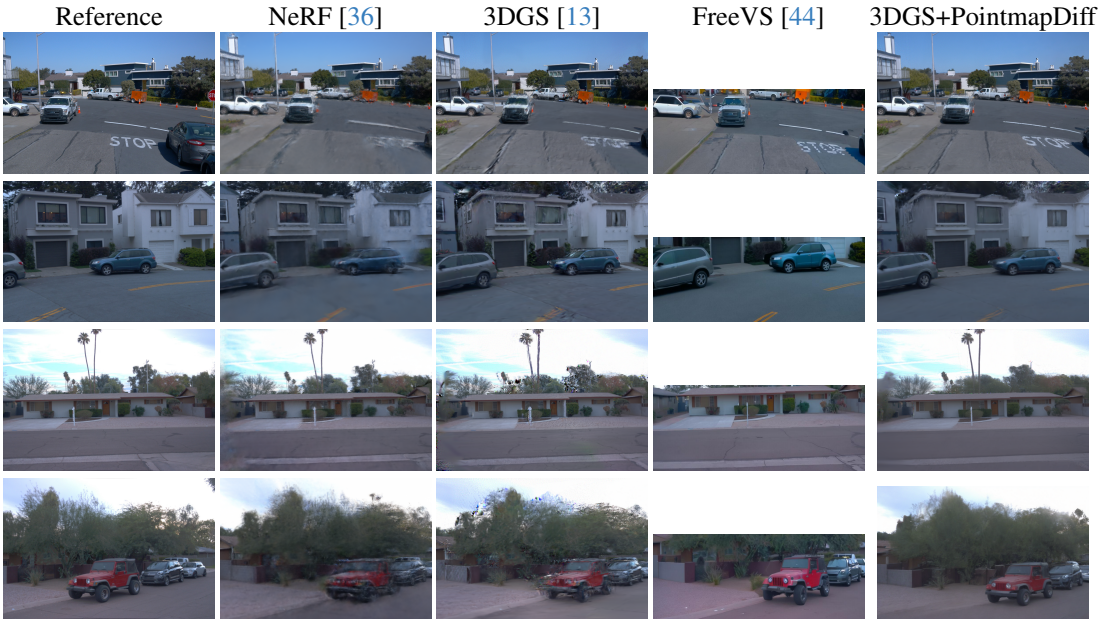


Figure 12. Addtional qualitative comparison on Waymo [34].

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

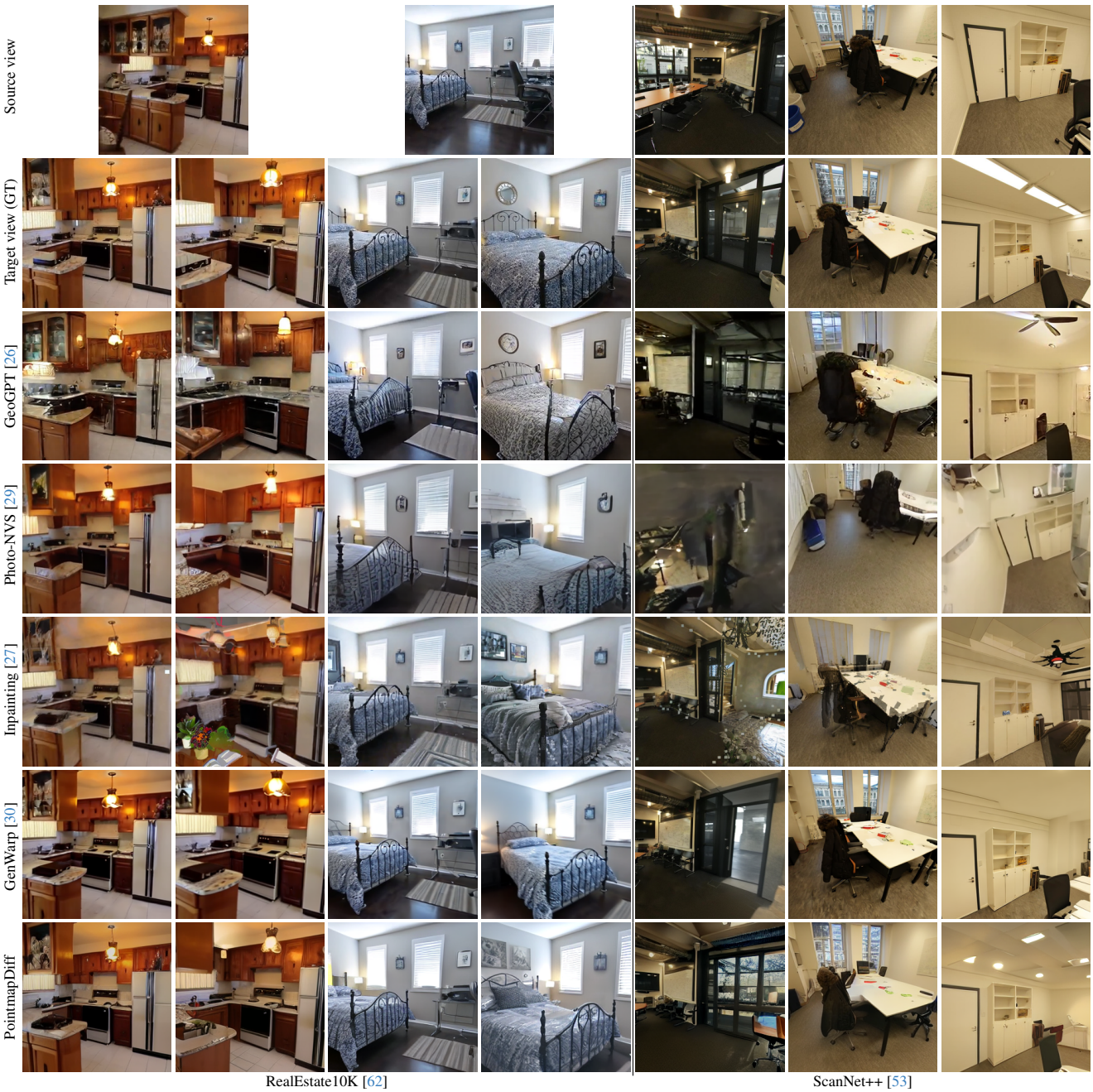


Figure 14. Additional NVS comparison on RealEstate10K [62] and ScanNet++ [53].

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.

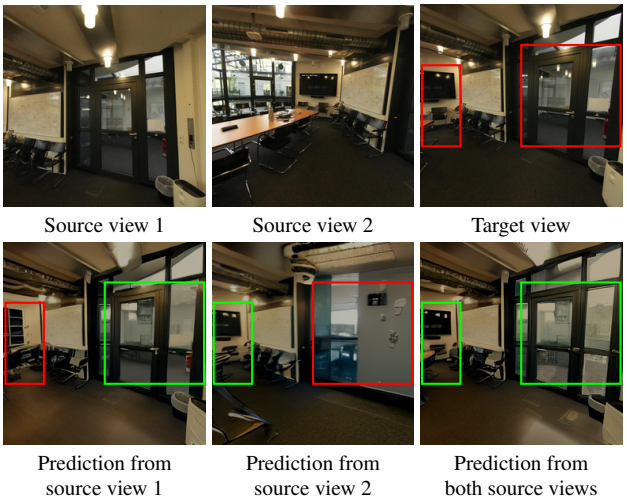


Figure 15. We demonstrate the results when generating viewpoints between two source views, effectively covering occluded regions by combining complementary information from both views. We use red to denote hallucinated regions and green to indicate aligned regions compared to the target view.

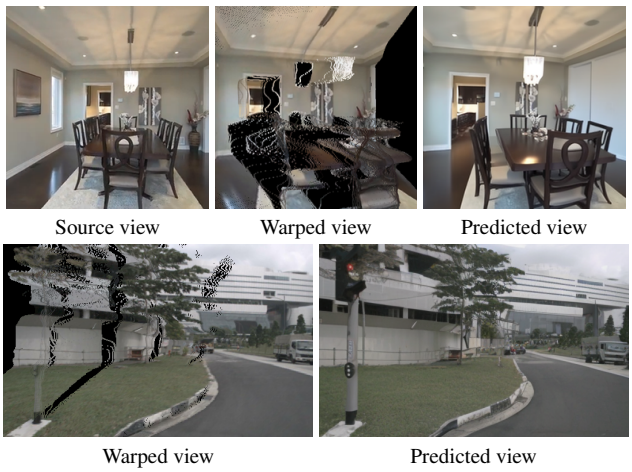
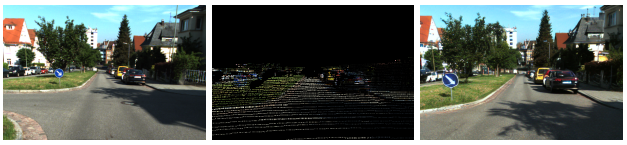


Figure 16. Robustness to noisy depth on RealEstate10K [62] and nuScenes [4].



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.