High-Fidelity Novel View Synthesis via Splatting-Guided Diffusion Supplementary Materials

XIANG ZHANG, ETH Zürich, Switzerland and DisneyResearch|Studios, Switzerland
YANG ZHANG, DisneyResearch|Studios, Switzerland
LUKAS MEHL, DisneyResearch|Studios, Switzerland
MARKUS GROSS, ETH Zürich, Switzerland and DisneyResearch|Studios, Switzerland
CHRISTOPHER SCHROERS, DisneyResearch|Studios, Switzerland

In the supplementary material, we first provide more implementation details in Sec. A and discuss the benefits of our design choices in Sec. B. Following that, we provide additional experiments in Sec. C, including robustness analysis and efficiency comparisons. More visual results, *e.g.*, in-the-wild samples and stereo video conversion, are also presented in Sec. D. Finally, we discuss the limitations and potential directions for future works in Sec. E.

CCS Concepts: • **Computing methodologies** → **Computational photography**; **3D imaging**.

Additional Key Words and Phrases: Novel view synthesis, pixel splatting, video diffusion model

ACM Reference Format:
Xiang Zhang, Yang Zhang, Lukas Mehl, Markus Gross, and Christopher Schroers. 2025. High-Fidelity Novel View Synthesis via Splatting-Guided Diffusion Supplementary Materials. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3721238.3730669

ALGORITHM 1: Texture Bridge Training

Input: pixel-splatted view $\mathbf{v}_{\text{tgt}}$, ground-truth target view $\mathbf{x}_{\text{tgt}}$
repeat
  $\mathbf{z}_{\text{tgt}} = \text{LatentEncoder}(\mathbf{x}_{\text{tgt}})$; `// Encode ground-truth view`
  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; `// Sample Gaussian noise`
  $\hat{\mathbf{z}}_{\text{tgt}} = \text{Diffusion}(\epsilon, \mathbf{z}_{\text{tgt}})$; `// Apply texture degradation`
  $\mathbf{f}^i_{\text{enc}} = \text{LatentEncoder}_i(\mathbf{v}_{\text{tgt}})$; `// Get splatting features`
  $\mathbf{f}^i_{\text{dec}} = \text{LatentDecoder}_i(\hat{\mathbf{z}}_{\text{tgt}})$; `// Get diffusion features`
  $\{\mathbf{f}^i_{\text{fuse}}\} = \text{TextureBridge}(\{\mathbf{f}^i_{\text{enc}}\}, \{\mathbf{f}^i_{\text{dec}}\})$;
  `// Feature fusion via texture bridge`
  $\hat{\mathbf{x}}_{\text{tgt}} = \text{LatentDecoder}(\{\mathbf{f}^i_{\text{fuse}}\})$; `// Decode novel view`
  Take gradient descent step based on $\mathcal{L}(\hat{\mathbf{x}}_{\text{tgt}}, \mathbf{x}_{\text{tgt}})$;
until *converged*;

Authors' addresses: Xiang Zhang, ETH Zürich, Zürich, Switzerland and DisneyResearch|Studios, Zürich, Switzerland, xiang.zhang@inf.ethz.ch; Yang Zhang, DisneyResearch|Studios, Zürich, Switzerland, yang.zhang@disneyresearch.com; Lukas Mehl, DisneyResearch|Studios, Zürich, Switzerland, lukas.mehl.-nd@disneyresearch.com; Markus Gross, ETH Zürich, Zürich, Switzerland and DisneyResearch|Studios, Zürich, Switzerland, grossm@inf.ethz.ch; Christopher Schroers, DisneyResearch|Studios, Zürich, Switzerland, christopher.schroers@disneyresearch.com.

SIGGRAPH Conference Papers '25, August 10–14, 2025, Vancouver, BC, Canada

ACM ISBN 979-8-4007-1540-2/2025/08
https://doi.org/10.1145/3721238.3730669

A MORE IMPLEMENTATION DETAILS

A.1 Texture Bridge Training

Algorithm 1 shows the complete training pipeline of the proposed texture bridge. In practice, the texture degradation technique can be pre-applied to generate degraded latents, *i.e.*, $\hat{\mathbf{z}}_{\text{tgt}}$, for the training dataset to accelerate the training process.

A.2 Training Datasets of Baselines

Tab. 1 lists the training datasets for all baselines in our paper. For fair comparisons, we mainly follow Flash3D [Szymanowicz et al. 2024] to conduct experiments on the RealEstate10K dataset, and we follow DepthSplat [Xu et al. 2024] for the evaluations on DL3DV-10K and DTU datasets. For the evaluation on the Spring dataset and in-the-wild samples, we use the same model trained on the DL3DV-10K dataset without additional fine-tuning. Benefiting from our texture bridge, SplatDiff is robust to varying resolutions, delivering stable performance across the DL3DV-10K dataset (256×448), the Spring dataset (512×768), and real-world samples (576×1024).

B DESIGN CHOICE DISCUSSIONS

B.1 Diffusion Conditioning

The choice of diffusion conditioning is crucial for achieving precise camera control and fine-grained details in the synthesized novel views. A commonly adopted approach is to use 3D correspondences, *e.g.*, 3D points in Diffusion as Shader [Gu et al. 2025], as conditioning. However, such methods often incur detail loss and require learning complex conditioning-to-color mappings. By contrast, our pixel-splatted views preserve fine-grained details and provide *explicit guidance* to video diffusion, boosting novel view synthesis performance as shown in Tab. 2.

B.2 Aligned Synthesis

To enable aligned synthesis for diffusion models, we generate the input view $\hat{\mathbf{v}}_{\text{tgt}}$ by our Training Pair Alignment (TPA) and Splatting Error Simulation (SES) and employ the aligned pairs $\{\hat{\mathbf{v}}_{\text{tgt}}, \mathbf{x}_{\text{tgt}}\}$ instead of $\{\mathbf{v}_{\text{tgt}}, \mathbf{x}_{\text{tgt}}\}$ for training. The key difference between $\hat{\mathbf{v}}_{\text{tgt}}$ and $\mathbf{v}_{\text{tgt}}$ is their *alignment* with ground-truth views. Generally, training with aligned pairs is important to achieve aligned synthesis, and having splatting errors during training helps a model learn to correct them. However, $\mathbf{v}_{\text{tgt}}$ is often too misaligned with the ground-truth views due to depth estimation errors, harming the learning process. Thus, we found it crucial to create aligned pairs using TPA *and* simultaneously introduce splatting errors through SES for high-quality novel view synthesis.

Table 1. List of training datasets. & indicates that datasets are used jointly for training. / means that different datasets are used to train separate models, depending on experimental settings. - means that the model is training-free.

Method	Main Task	Training Datasets
Syn-Sin	Single-view NVS	RealEstate10K [Zhou et al. 2018]
SV-MPI	Single-view NVS	RealEstate10K [Zhou et al. 2018]
BTS	Single-view NVS	KITTI [Geiger et al. 2013] & KITTI-360 [Liao et al. 2022] & RealEstate10K [Zhou et al. 2018]
Splatter Image	Single-view NVS	RealEstate10K [Zhou et al. 2018]
MINE	Single-view NVS	RealEstate10K [Zhou et al. 2018]
AdaMPI	Single-view NVS	COCO [Caesar et al. 2018]
Flash3D	Single-view NVS	RealEstate10K [Zhou et al. 2018]
GenWarp	Single-view NVS	RealEstate10K [Zhou et al. 2018] & ACID [Liu et al. 2021] & ScanNet [Dai et al. 2017]
ViewCrafter	Single-view NVS	RealEstate10K [Zhou et al. 2018] & DL3DV-10K [Ling et al. 2024]
Diffusion as Shader	Single-view NVS	MiraData [Ju et al. 2024]
NVS-Solver	Single-view NVS	-
DepthSplat	Sparse-view NVS	RealEstate10K [Zhou et al. 2018] / DL3DV-10K [Ling et al. 2024]
pixelSplat	Sparse-view NVS	RealEstate10K [Zhou et al. 2018]
MVSplat	Sparse-view NVS	RealEstate10K [Zhou et al. 2018] / DL3DV-10K [Ling et al. 2024]
TranSplat	Sparse-view NVS	RealEstate10K [Zhou et al. 2018]
StereoCrafter	Stereo Conversion	Private stereo datasets
SplatDiff (Ours)	All	RealEstate10K [Zhou et al. 2018] / DL3DV-10K [Ling et al. 2024]

B.3 Texture Bridge

Latent diffusion models are widely used to balance the performance and complexity [Rombach et al. 2022; Xing et al. 2025]. However, texture details are often compressed by the latent encoder during pixel-to-latent conversion, making the recovery of high-fidelity texture ill-posed. Rather than designing complex texture synthesis modules in latent space (*e.g.*, ControlNet [Zhang et al. 2023]), our texture bridge directly re-uses the rich features from the latent encoder, allowing simple layers to achieve remarkable performance (*e.g.*, see Tab. 2).

B.4 Texture Degradation

The goal of our texture bridge is to adaptively fuse the splatted view and the diffusion output for high-quality view synthesis. As depicted in Fig. 1, the splatted view shows better textures (green box) but contains unknown regions and splatting errors. Although the diffusion output generates reasonable contents for the unknown regions (red box), they often suffer from texture hallucination (green box). Ideally, the texture bridge should learn to detect the problematic regions, *e.g.*, splatting errors and hallucinated textures, and adaptively utilize better features for synthesis. However, directly using the diffusion output often leads to sub-optimal training performance mainly due to the inconsistent contents in the unknown regions (red box in Fig. 1). To address this, we propose the texture degradation strategy and generate the degraded views for training. As shown in Fig. 1, the degraded view shares similar contents with the target view while imitating the texture hallucination effects in real diffusion outputs. By substituting the diffusion output with the degraded view for training, our texture bridge better learns to rely more on the generated contents for the unknown regions while maintaining the ability to detect hallucinated textures.

C ADDITIONAL EXPERIMENTS

C.1 Model Robustness

View transformation errors (*e.g.*, distortions or misalignments) and large viewpoint shifts pose significant challenges to novel view synthesis. To test the robustness of our SplatDiff, we add an additional super hard set on the DL3DV-10K dataset and introduce two metrics to evaluate the difficulty of each setting:

- **Relative Splatting Error (RSE)**:

$$\mathrm{RSE} = \frac{1}{N}\sum_{k=1}^{N}\frac{|\mathbf{x}_k^{\mathrm{splat}} - \mathbf{x}_k^{\mathrm{gt}}|}{\mathbf{x}_k^{\mathrm{gt}}}$$

  where N is the number of valid splatted pixels, and $\mathbf{x}_k^{\mathrm{splat}}, \mathbf{x}_k^{\mathrm{gt}}$ correspond to the pixel values in splatted and ground-truth views. Higher RSE reflects more view transformation errors.
- **Valid Splatting Ratio (VSR)**:

$$\mathrm{VSR} = \frac{N}{N_{\mathrm{total}}},$$

  with N_{total} being the total pixels in splatted views. Lower VSR implies more disocclusions and larger viewpoint shifts.

In Tab. 2, we perform comparisons across easy, hard, and super hard sets with varying RSE and VSR levels (visual examples for each set can be found in Fig. 2). Due to the large disocclusion regions in the input images, we follow previous works to compute pixel-level metrics PSNR and SSIM only on the valid splatting regions (denoted by mPSNR and mSSIM). Tab. 2 highlights the strong performance of SplatDiff not only in common but also in challenging scenarios.

Fig. 1. **Example of texture degradation.** Although the original diffusion output fills the unknown regions in the splatted view, the generated contents usually differ from the target view (red box), resulting in sub-optimal training performance. Thus, we employ the degraded view for training, which shares similar contents with the target view while imitating the texture hallucination effects (green box) in real diffusion outputs. Images credited to [Ling et al. 2024].

Table 2. **Performance comparisons on DL3DV-10K dataset under varying levels of viewpoint shifts and transformation errors**. The Relative Splatting Error (RSE) and Valid Splatting Ratio (VSR) metrics are introduced to measure the difficulty of each setting (Higher RSE reflects more view transformation errors, and lower VSR implies more disocclusions and larger viewpoint shifts). To exclude the disoccluded regions in pixel-level metrics, we compute masked PSNR and SSIM (denoted by mPSNR and mSSIM) only on the valid splatting regions. * denotes methods with two input views. Best and second-best results are marked.

Method	Easy (RSE=0.565, VSR=91.98%)					Hard (RSE=0.652, VSR=78.86%)					Super Hard (RSE=0.744, VSR=64.16%)				
	mPSNR	mSSIM	LPIPS	DISTS	FID	mPSNR	mSSIM	LPIPS	DISTS	FID	mPSNR	mSSIM	LPIPS	DISTS	FID
Diffusion as Shader	16.80	0.471	0.363	0.150	62.07	15.20	0.442	0.457	0.190	89.55	14.47	0.451	0.507	0.219	108.11
NVS-Solver	19.58	0.644	0.309	0.155	81.68	17.29	0.590	0.374	0.183	100.42	15.50	0.543	0.437	0.201	112.59
ViewCrafter	21.06	0.684	0.182	0.349	46.60	19.43	0.651	0.249	0.349	59.56	18.39	0.640	0.321	0.357	72.77
DepthSplat*	22.89	0.801	0.168	0.109	52.01	21.79	0.773	0.221	0.124	61.37	20.94	0.758	0.280	0.140	73.60
SplatDiff (Ours)	26.27	0.865	0.113	0.068	24.23	23.18	0.810	0.181	0.092	41.22	21.53	0.781	0.252	0.113	54.19

Fig. 2. **Visual examples of easy, hard, and super hard sets on the DL3DV-10K dataset.** Images credited to [Ling et al. 2024].

C.2 Impact of Texture Degradation

Texture degradation is designed to improve the feature selection ability of our texture bridge. Significant deviation between diffusion outputs and ground-truth views in disoccluded regions often misguides the texture bridge's feature selection during training. In contrast, the degraded views align better with the ground-truth in disoccluded regions, encouraging the texture bridge to leverage and improve diffusion features for reconstruction. To verify this

Table 3. **Impact of texture degradation.** Best results are marked.

Method	Super Hard Set on DL3DV-10K				
	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	FID ↓
w/o texture degradation	20.00	0.655	0.261	0.123	60.94
w/ texture degradation	20.30	0.658	0.252	0.113	54.19

Table 4. **Texture-bridge-only ablation.** Best and second-best results are marked.

ID	Texture Bridge	Aligned Synthesis	DL3DV-10K Dataset				
			PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	FID ↓
A-1			23.21	0.694	0.182	0.349	46.60
A-2	✓		25.56	0.824	0.142	0.077	31.89
A-3	✓	✓	26.05	0.825	0.118	0.070	25.49

Splatted View Model A-2 Model A-3

Fig. 3. **Example results of texture-bridge-only model.** The model ID is consistent with Tab. 4, where model A-2 only uses texture bridge, and model A-3 combines texture bridge and aligned synthesis. Images credited to [Ling et al. 2024].

point, we perform experiments on the super hard set on DL3DV-10K, which contains large disoccluded regions, and Tab. 3 confirms the benefits of texture degradation.

C.3 Texture-Bridge-Only Ablation

By utilizing splatting features from the latent encoder, the texture-bridge-only model (model A-2) achieves promising quantitative results as shown in Tab. 4. However, it often struggles with unnatural transitions at disocclusion boundaries, due to inconsistent color/geometry between pixel-splatted views and diffusion outputs (*e.g.*, see Fig. 3). Our aligned synthesis method significantly improves the consistency between the splatted view and the diffusion outputs, leading to better view synthesis performance as shown in Tab. 4 and Fig. 3.

C.4 Importance of Video Diffusion Prior

Compared with image diffusion models, the video prior in video diffusion models benefits novel view synthesis tasks, as synthesizing novel views can be regarded as generating videos captured around target scenes. To verify this, we introduce an Inter-view Alignment Error (IAE) to measure the inter-view consistency:

$$\text{IAE} = \frac{1}{2(T-1)} \sum_{k=1}^{T-1} \Big(\sigma(\text{Warp}(\hat{\mathbf{x}}_k, \mathbf{m}_k^{k+1}), \hat{\mathbf{x}}_{k+1}) + \sigma(\text{Warp}(\hat{\mathbf{x}}_{k+1}, \mathbf{m}_{k+1}^{k}), \hat{\mathbf{x}}_k) \Big),$$

where $T, \hat{\mathbf{x}}_k, \mathbf{m}_k^{k+1}$ are the number of synthesized views, the k-th synthesized view, and the transformation from the k-th view to k+1-th view, respectively. We use masked PSNR and SSIM as $\sigma(\cdot)$ (denoted by IAE-PSNR and IAE-SSIM). Tab. 5 demonstrates the better consistency of video diffusion. SplatDiff also outperforms ViewCrafter by addressing texture/geometry hallucination.

Table 5. **Inter-view consistency.** We use masked PSNR and SSIM to compute the inter-view alignment error, which are denoted by IAE-PSNR and IAE-SSIM, respectively. Best and second-best results are marked.

Method	Diffusion Type	IAE-PSNR ↑	IAE-SSIM ↑
GenWarp	Image Diffusion	15.30	0.365
ViewCrafter	Video Diffusion	21.71	0.662
SplatDiff (Ours)	Video Diffusion	25.15	0.786

Table 6. **Efficiency comparisons.** We compare the peak GPU memory, the number of model parameters, and the inference speed on the DL3DV-10K dataset with a NVIDIA GeForce RTX 4090 GPU. Note that only the diffusion part is taken into account for comparisons. * uses NVIDIA A100 GPU due to out-of-memory. Best and second-best results are marked.

Method	Peak GPU Mem.	#Parameters	Speed
Diffusion as Shader*	28.44 G	13.11 B	378.09 s
NVS-Solver	21.60 G	2.25 B	100.37 s
ViewCrafter	16.30 G	2.22 B	26.66 s
SplatDiff (Ours)	17.69 G	2.28 B	26.75 s

C.5 Efficiency Comparisons

We compare the peak GPU memory, the number of model parameters, and the inference speed of our SplatDiff with recent state-of-the-art NVS approaches. Combining Tab. 2 and Tab. 6, SplatDiff achieves state-of-the-art performance while maintaining competitive efficiency.

D MORE VISUAL RESULTS

D.1 Visual Comparisons in Ablation Study

Fig. 4 provides visual results for the ablation study (Tab. 2 in the main paper). We also compute the absolute difference map between the splatted views and the corresponding novel views to visualize the difference regions. It is obvious that the baseline model synthesizes misaligned novel views and produces hallucinated textures, leading to significant difference regions across the entire image. With the aligned synthesis strategy, the model (#3 in Tab. 2 of the main paper) generates geometry-consistent results but still suffers from texture hallucination as shown in the green box of Fig. 4. By contrast, our SplatDiff utilizes the information from the splatted view with

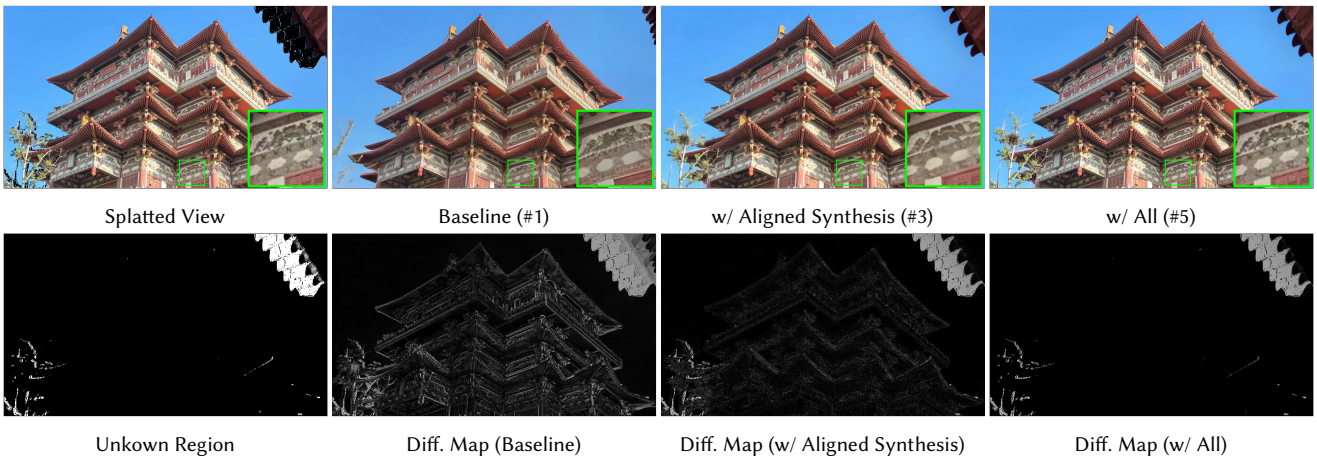


Fig. 4. **Visual comparisons of ablation study.** The difference map shows the absolute difference between the splatted view and the corresponding novel view. The model ID is consistent with Tab. 2 in the main paper. The baseline model generates a misaligned novel view with the conditioned splatted view, showing significant differences across the image. With the proposed aligned synthesis strategy, model #3 better follows the conditioning but still suffers from texture hallucination (green box). Combining the aligned synthesis and the texture bridge, our model (#5) synthesizes geometry-aligned novel views while recovering high-fidelity texture.

the proposed texture bridge, achieving high-fidelity novel view synthesis.

D.2 Zero-Shot Performance

To verify the zero-shot performance of SplatDiff, we provide additional visual comparisons on diverse in-the-wild samples, including landscapes, buildings, animals, and paintings (Fig. 5). Previous approaches, *e.g.*, ViewCrafter [Yu et al. 2024], often synthesize contents that differ from the inputs, like the hallucinated lighting and the rooftop texture in the first-row example of Fig. 5. In contrast, our method preserves the geometric layout while recovering fine-grained details as shown in the green box of Fig. 5. In addition, benefiting from the aligned synthesis strategy, our method corrects the splatting errors and fills reasonable contents for the unknown regions, *e.g.*, the green boxes in the bird and the axe examples depicted in Fig. 5.

D.3 Video Results in Stereo Conversion

In Fig. 6, we provide more stereo video conversion results on the Spring dataset [Mehl et al. 2023]. Since ViewCrafter is trained only on static scenes, it is challenging for it to handle dynamic inputs, resulting in severe content hallucination as shown in Fig. 6. Although our method is trained on the same static datasets as ViewCrafter, SplatDiff exhibits promising zero-shot performance in dynamic scenes. This is because our texture bridge module leverages the splatted views to maintain the same motion as the input video. Meanwhile, with our aligned synthesis strategy, the diffusion model performs similarly to inpainting models and fills reasonable contents for the unknown regions, *e.g.*, disocclusions in the last two rows of Fig. 6a, achieving high-quality synthesis of the stereo video.

Compared with StereoCrafter [Zhao et al. 2024] which is designed specifically for stereo video conversion, our method still shows superior performance in synthesis quality and consistency. As depicted in Fig. 6a, StereoCrafter tends to produce novel views with smoothed details, *e.g.*, the rocks on the road, and fills blurred contents for the disocclusion regions. For consistency, while StereoCrafter generates better novel views than ViewCrafter, inconsistent details are often observed in the outputs of StereoCrafter, such as the girl's eyes in the green box of Fig. 6b. Compared with previous approaches, our method produces the best stereo conversion results with consistent geometry and realistic details, verifying the effectiveness and versatility of SplatDiff.

E LIMITATIONS AND FUTURE WORKS

While our SplatDiff achieves remarkable performance on many novel view synthesis tasks, limitations still exist: (i) **View-Dependent Effects**: Compared with the popular Gaussian splatting techniques, SplatDiff has demonstrated the effectiveness of the pixel splatting method under limited input views. However, since current pixel splatting methods generally assume the brightness constancy across different viewpoints, how to handle the view-dependent effects, *e.g.*, reflective surfaces, remains an open problem (*e.g.*, see failure case in Fig. 7a). Although Gaussians are capable of modeling view-dependent effects, estimating accurate Gaussian parameters from limited observations is challenging. One potential solution is to leverage the rich prior from foundation models to facilitate the modeling of view-dependent effects, and we leave it for future works. (ii) **Long-Range Consistency**: Most existing video diffusion models are designed to output a pre-defined number of images, and thus multiple inferences are required to handle long-range inputs, *e.g.*, long videos in stereo conversion. However, due to the generative nature of diffusion models, the synthesized contents are usually different in multiple inferences. Although our approach utilizes the texture bridge to maintain the consistency on the splatted regions, the contents on the unknown regions, *e.g.*, disocclusions, might

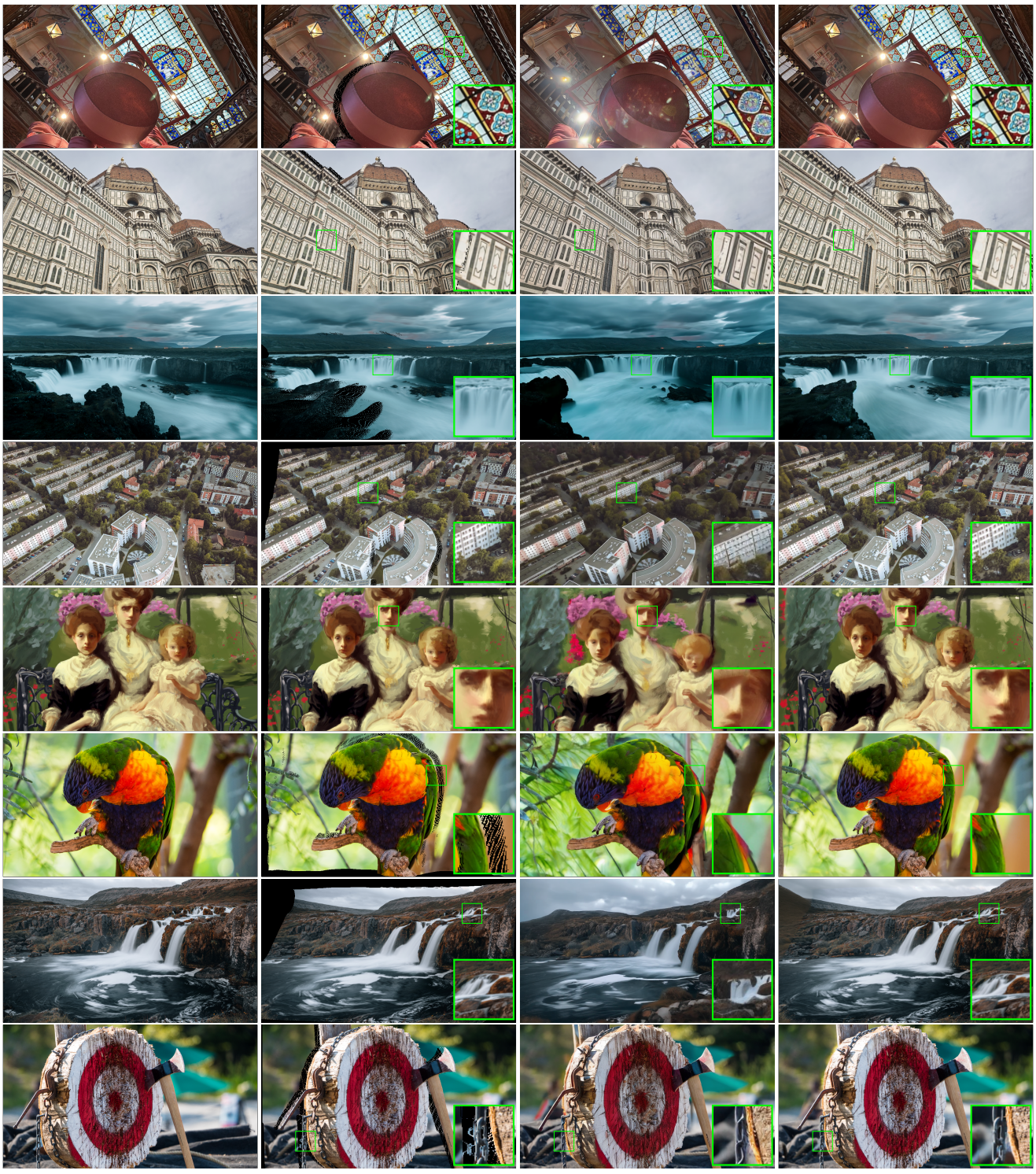
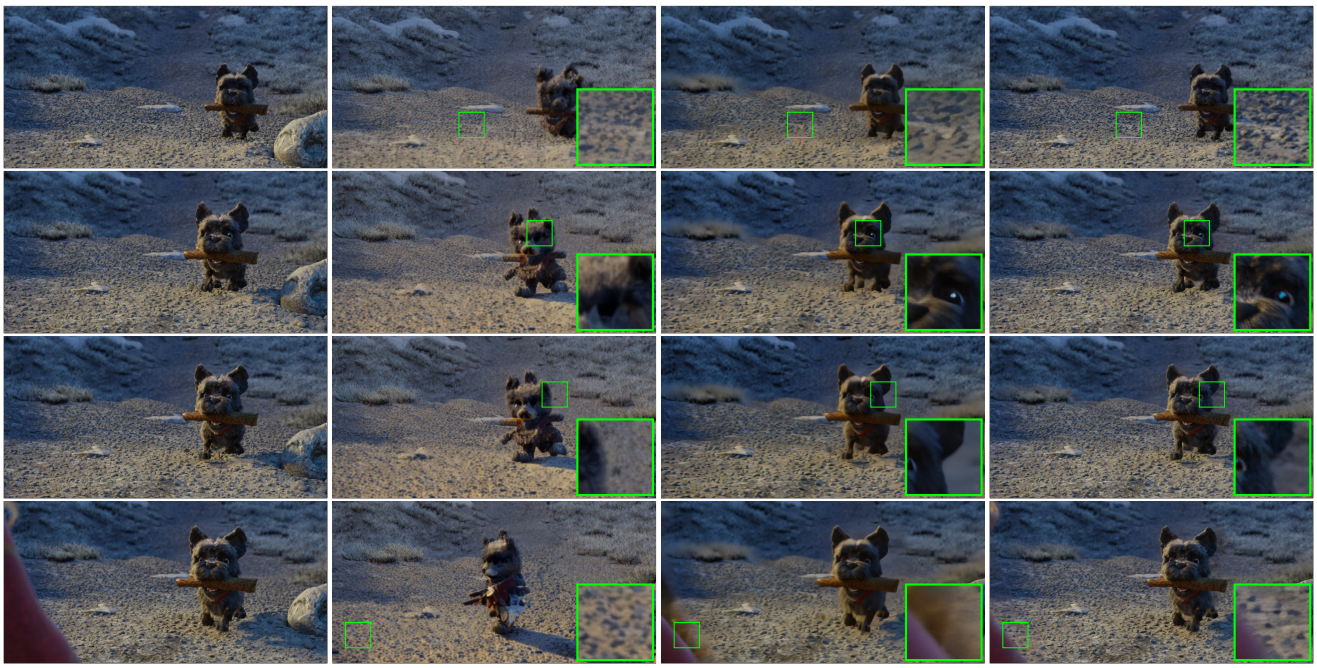


Fig. 5. **Qualitative comparisons on in-the-wild high-resolution (576×1024) samples.**



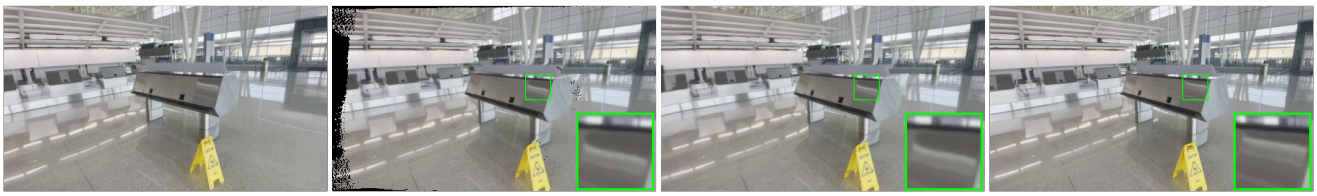
Input Views (Left) ViewCrafter StereoCrafter **SplatDiff (Ours)**

(a) Visual comparisons on details

Input Views (Left) ViewCrafter StereoCrafter **SplatDiff (Ours)**

(b) Visual comparisons on consistency

Fig. 6. **Stereo video conversion results on the Spring dataset.** Right-eye views are synthesized based on the input left-eye views. Images credited to [Mehl et al. 2023].



Input View | Splatted View | Synthesized Novel View | Target View

(a) View-dependent effects

Frame 20 in the first inference | Frame 25 in the first inference | Frame 5 in the second inference | Frame 10 in the second inference

(b) Long-range consistency

Fig. 7. **Visual examples of failure cases.** (a) Due to the assumption of brightness constancy in pixel splatting, our method can fail to model view-dependent effects, *e.g.*, the reflective surface in the green box. (b) Since the number of output frames is set to 25 in our experiments, our method might fail to generate consistent content in disoccluded regions for long videos with more than 25 frames. Images credited to [Ling et al. 2024] and [Mehl et al. 2023].

differ. As the example shown in Fig. 7b, although our model maintains strong consistency within each inference (*e.g.*, Frame 20 and 25 in the first inference), different contents might be generated for the disoccluded region in different inferences (*e.g.*, Frame 25 in the first inference *v.s.* Frame 5 in the second inference). To achieve long-range consistency, one could draw inspiration from the recent video-based methods, *e.g.*, rolling inference [Ke et al. 2024], for improved consistency of diffusion outputs, and combine the techniques in our SplatDiff for high-fidelity view synthesis.

REFERENCES

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. 2018. Coco-stuff: Thing and stuff classes in context. In *CVPR*. 1209–1218.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5828–5839.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. 2013. Vision meets robotics: The kitti dataset. *IJRR* 32, 11 (2013), 1231–1237.

Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, et al. 2025. Diffusion as Shader: 3D-aware Video Diffusion for Versatile Video Generation Control. *arXiv preprint arXiv:2501.03847* (2025).

Xuan Ju, Yiming Gao, Zhaoyang Zhang, Ziyang Yuan, Xintao Wang, Ailing Zeng, Yu Xiong, Qiang Xu, and Ying Shan. 2024. Miradata: A large-scale video dataset with long durations and structured captions. *NeurIPS* 37 (2024), 48955–48970.

Bingxin Ke, Dominik Narnhofer, Shengyu Huang, Lei Ke, Torben Peters, Katerina Fragkiadaki, Anton Obukhov, and Konrad Schindler. 2024. Video Depth without Video Models. *arXiv preprint arXiv:2411.19189* (2024).

Yiyi Liao, Jun Xie, and Andreas Geiger. 2022. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE TPAMI* 45, 3 (2022), 3292–3310.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. 2024. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*. 22160–22169.

Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. 2021. Infinite nature: Perpetual view generation of natural scenes from a single image. In *ICCV*. 14458–14467.

Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. 2023. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *CVPR*. 4981–4991.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*. 10684–10695.

Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, João F Henriques, Christian Rupprecht, and Andrea Vedaldi. 2024. Flash3D: Feed-Forward Generalisable 3D Scene Reconstruction from a Single Image. *arXiv preprint arXiv:2406.04343* (2024).

Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. 2025. Dynamicrafter: Animating open-domain images with video diffusion priors. In *ECCV*. Springer, 399–417.

Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. 2024. Depthsplat: Connecting gaussian splatting and depth. *arXiv preprint arXiv:2410.13862* (2024).

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

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *ICCV*. 3836–3847.

Sijie Zhao, Wenbo Hu, Xiaodong Cun, Yong Zhang, Xiaoyu Li, Zhe Kong, Xiangjun Gao, Muyao Niu, and Ying Shan. 2024. Stereocrafter: Diffusion-based generation of long and high-fidelity stereoscopic 3d from monocular videos. *arXiv preprint arXiv:2409.07447* (2024).

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. 2018. Stereo magnification: learning view synthesis using multiplane images. *ACM Trans. Graph.* 37, 4, Article 65 (July 2018), 12 pages. https://doi.org/10.1145/3197517.3201323