Restereo: Unifying diffusion stereo video generation and restoration

Xingchang Huang[1,2] Ashish Kumar Singh[4] Florian Dubost[4] Cristina Nader Vasconcelos[5]
Sakar Khattar[4] Liang Shi[4] Christian Theobalt[1,2] Cengiz Öztireli[3,4] Gurprit Singh[1,2]

[1]Max Planck Institute for Informatics [2]VIA-Center Saarbücken
[3]University of Cambridge [4]Google [5]Google DeepMind

Abstract

Stereo video generation has gained increasing interest, yet most video diffusion models remain limited to monocular outputs. Existing stereo approaches struggle with degraded inputs and focus on stereo generation only. In this paper, we propose a diffusion framework that jointly performs stereo video generation and restoration from degraded video. The key idea is to inject video degradations during training and condition the model on warped masks, allowing it to learn robust two-view consistency from data, even though explicit view-consistency losses are not feasible in latent diffusion models. This design enables fine-tuning on synthetic data and allows a single model to handle different degradation types, including downsampling and noise, while generalizing to real-world scenarios. Experiments show that our method outperforms existing baselines across various degradation levels, while matching their performance on non-degraded inputs and achieving a superior quality–time tradeoff.

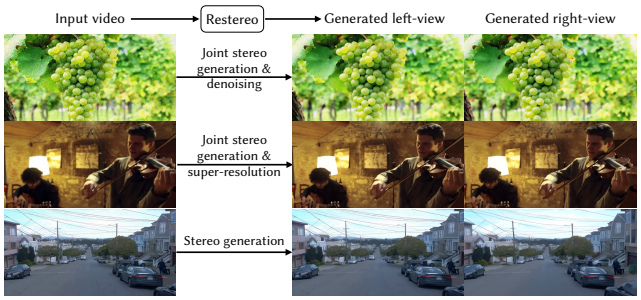

Figure 1. Our method, called Restereo, is fine-tuned to be a unified model that performs joint and robust stereo video generation and restoration under various levels of degradation (noise, downsampling, and no degradation). Input videos above are from Pixabay [6], DAVIS [7], and Waymo Open Dataset [8], respectively. The quality of all stereo videos in the paper can also be examined by cross-eye viewing given the left- and right-images, flipping the left and right views in an image viewer, or through the Supplemental HTML videos compatible with a VR headset.

1. Introduction

Stereo video generation is becoming increasingly crucial for creating immersive experiences in modern VR applications. Given the strength of pretrained video diffusion models, recent work has explored training-free stereo video generation using pretrained diffusion models [1]. Some heuristic-based training-free methods [2–5] do not require training on large datasets, making them lightweight and easy to use. However, they rely solely on pretrained diffusion models to inpaint the right-view latent space, which can result in spatial and temporal inconsistencies due to the lack of latent stereo-specific training. As a result, the decoded right-view images may misalign with the left view and exhibit artifacts, especially around occluded regions.

Training-based methods [9–12] have shown promising results in stereo video generation. These methods are designed to train on a large-scale dataset of stereo videos and have become foundation models in stereo video generation. However, these datasets, though diverse, are costly to collect and annotate, and are not publicly available.

Moreover, existing methods focus on inpainting left-warped videos to generate the right view, limiting them to stereo generation only. They do not handle degraded inputs or enhance both views jointly, resulting in low-quality stereo videos given degraded input. An alternative approach is to use a two-stage pipeline that first restores the degraded input video and then generates the right view. However, compared to our unified model, this setup can suffer from distribution mismatch between the restored inputs and the stereo generation model's training data, limited robustness to varying degradation levels depending on how the restoration model is trained, as well as increased computational and parameter overhead.

In this work, we design a training/fine-tuning pipeline that injects video degradations, enabling the model to learn from synthetic data and robustly recover stereo-consistent

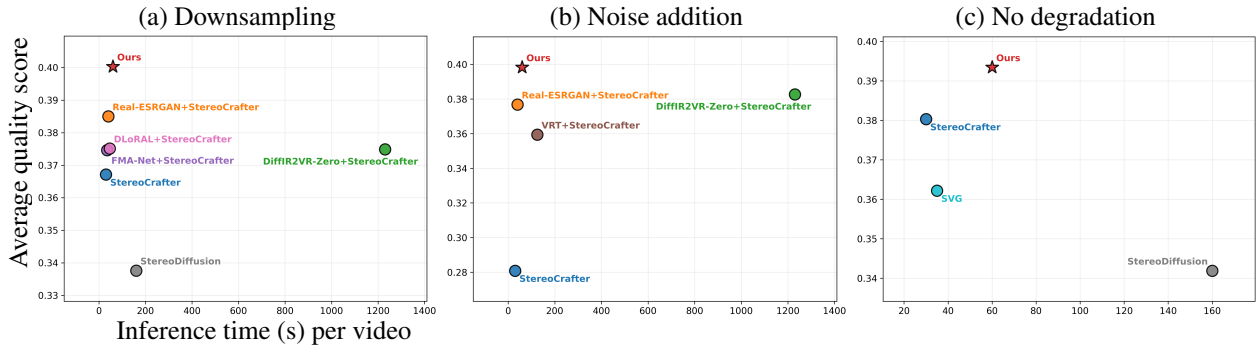

Figure 2. This figure shows inference time and average stereo quality score evaluated on our created test data. The score is computed as the average of view- and temporal-consistency scores in Table 1. The results indicate that: (1) under downsampling and noise degradation on input video, our method can generate more consistent stereo pairs compared to applying a video restoration method (Real-ESRGAN [13], DLoRAL [14], FMA-Net [15], DiffIR2VR-Zero [16], VRT [17]) first followed by a stereo video generation method (e.g., StereoCrafter [9]) (2) without degradation, our method shows comparable quality-time tradeoff to state-of-the art methods (StereoCrafter [9], SVG [3], StereoDiffusion [4]).

pairs. In summary, we make the following contributions:

- a data augmentation strategy that injects video degradations, enabling the fine-tuned stereo model to jointly restore and generate stereo video from degraded input videos (as shown in Figure 1),
- a synthetic-driven pipeline for stereo video models that removes the need for complex stereo preprocessing while generalizing to real-world videos with various degradation types.

Both the code and data will be made publicly available on the project website.

2. Related work

Diffusion-based stereo generation. Training-free diffusion-based stereo generation models, such as StereoDiffusion [4] and SVG [3], apply warping in the latent space of pretrained diffusion models. However, its reliance on latent inpainting limits quality and scalability to videos. Training-based methods like StereoCrafter [9], SpatialMe [10], Text2Stereo [18] achieve strong spatial and temporal consistency. A recent work on precise camera control in multi-view video diffusion, GEN3C [19], is closely related to the stereo setup. However, training-based methods require large stereo video datasets and complex preprocessing, and they assume high-quality inputs and can fail drastically to generate consistent stereo pairs when the input videos are degraded. In contrast, we adopt a lightweight training-based approach using synthetic data and account for degraded video input for more robust generation.

Video restoration. An alternative to stereo generation and restoration in one stage, video restoration models can be used to enhance degraded videos before applying stereo generation models. Real-ESRGAN [13] proposes to train image super-resolution models via degrading synthetic data. VRT [17] is a novel video restoration transformer architecture for temporally coherent video super-resolution, deblurring, and denoising. Diffusion-based methods [14, 16, 20] use generative diffusion priors to improve generalization but can demand high GPU memory. Flow-guided models also enforce spatial and temporal consistency, with FMA-Net [15] advancing joint video super-resolution and deblurring through a specialized flow-guided module design.

We compare against most state-of-the-art video restoration methods, but their restoration quality does not reliably transfer to stereo generation. Using them as a preprocessing stage often introduces inconsistencies and artifacts due to the lack of joint optimization. While our method may not be state-of-the-art in restoration alone, it produces more stereo-consistent pairs, as reflected in the average stereo quality scores in Figure 2.

3. Method

3.1. Overview

Given an input left-view video represented by $\boldsymbol{x}^{(l)}$ with F =16 frames, our goal is to generate enhanced left-view video $\hat{\boldsymbol{x}}^{(l)}$ and right-view video $\hat{\boldsymbol{x}}^{(r)}$. Unlike prior methods [3, 4, 9], we target joint stereo video generation and restoration where the input video may be degraded, and both views require enhancement. To this end, our pipeline trains both left-to-right and left-to-left generation branches, as shown in Figure 4. Although we fine-tune from a pretrained model rather than train from scratch, our approach is not limited to fine-tuning, and we use the terms interchangeably throughout the paper.

3.2. Stereo video data generation

First, we need to generate training data for training our proposed model. We use Kubric [21], a graphics data generation pipeline, to generate synthetic training data. This allows us to create 3D scenes with preprocessed ShapeNet [22] objects, environment lighting [23], camera positions, and provide ground truth data like depth maps.

Data augmentation with video degradation. A key step is to inject different types of video degradation during training. We augment the dataset by applying different degradation settings: downsampling, Gaussian noise, a combination of both, or no degradation with random strengths during training. Regarding resolution, training and testing videos are all with spatial resolution 1024×512, preprocessed with the following equation:

$$\boldsymbol{x}' = \texttt{Up}(\texttt{Down}(\boldsymbol{x})) \tag{1}$$

Here we omit the superscript of $\boldsymbol{x}$, but $\boldsymbol{x}$ and $\boldsymbol{x}'$ can represent both left ($\boldsymbol{x}^{(l)}$, $\boldsymbol{x}'^{(l)}$) and right ($\boldsymbol{x}^{(r)}$, $\boldsymbol{x}'^{(r)}$) view videos consistent to the notations in Figure 4. Each frame of the video $\boldsymbol{x}$ is downsampled with the `Down` operator to $W' \times H'$ implemented in PyTorch with:

$$\texttt{Interp} = \texttt{F.interpolate}(x,\ \text{size} = (W', H'),\ \text{mode} = \text{'area'}) \tag{2}$$

During training, $W' \times H' = 256 \times 128$ or 512×256 or 1024×512. `Up` is the same as `Interp` to turn the degraded video back to 1024×512. Gaussian noise is applied with a standard deviation ranging from 1 to 100. During testing, $\boldsymbol{x}_i$ is downsampled (`Down`) to 256×128, 320×160 or 360×180 and then restored back to 1024×512 via `Up` refers to video restoration method to get $\boldsymbol{x}'$. Gaussian noise is applied with a standard deviation of 30. $\boldsymbol{x}'$ is obtained after restoration and the input to a stereo generation method such as StereoCrafter [9].

The data augmentation process is inspired by Real-ESRGAN [13], which trains a generator for image restoration by degrading synthetic images. But their designed degradation steps are more tailored to anime videos. We show that data degradation is crucial for joint generation and restoration in Figure 3. Our generated right-view video with augmentation contains sharper details around the edges than the one without augmentation. We show training examples in Figure 4 and in Supplemental Figure 2.

3.3. Video warping

We follow the StereoCrafter [9] warping pipeline but reimplement forward warping to match our synthetic stereo generation. Given the left-view video and depth maps, we compute pixel displacements using inverse depth scaled by a disparity factor S. Overlapping pixels are resolved using a z-buffer, and unassigned pixels in the right view are filled via bilinear interpolation to remove flying pixels.

During training, ground-truth depth maps are provided by the graphics engine. During inference, we use DepthCrafter [25] for video depth estimation. Due to depth prediction imperfections, the warped masks may contain artifacts. We therefore post-process the masks with morphological dilation to remove flying pixels and fill holes in occluded regions. We also experimented with DepthPro [26], but observed lower temporal consistency compared to DepthCrafter.

3.4. Fine-tuning

At each training/fine-tuning iteration, we randomly sample a branch as shown in Figure 4 left-most column. For the left-to-right branch, input consists of a left-view input video $\boldsymbol{x}^{(l)} \in \mathbb{R}^{F \times H \times W \times 3}$, a warped video $\boldsymbol{x}^{(w)} \in \mathbb{R}^{F \times H \times W \times 3}$ based on disparity-based forward mapping, and a binary warped mask $\boldsymbol{M}$ representing the occluded region after warping. $\boldsymbol{x}^{(w)}$ is then degraded via our data augmentation to be $\boldsymbol{x}'^{(w)}$. Then $\boldsymbol{x}'^{(w)}$ is compressed via a frozen VAE encoder to latents $\boldsymbol{z}'^{(w)} \in \mathbb{R}^{F \times H' \times W' \times 4}$ with a factor of 8 ($H' = H/8, W' = W/8$). $\boldsymbol{M}$ is resized to $\boldsymbol{M}$' $\in \mathbb{R}^{F \times H' \times W' \times 1}$, with nearest-neighbor interpolation to ensure that the mask's binary nature is preserved. Diffusion models also take as input the target video added with Gaussian noise, represented as $\boldsymbol{z}_t^{(r)}$, where t is a timestep of a diffusion forward process. $\boldsymbol{z}_t^{(r)}$, $\boldsymbol{M}$' and $\boldsymbol{z}'^{(w)}$ are concatenated along the channel dimension, which serves as the input to the Diffusion U-Net [1]. Thus, we extend the U-Net input layer with an additional zero-initialized channel so that the initial output remains unchanged before fine-tuning. The output of Diffusion U-Net is $\hat{\boldsymbol{z}}^{(r)}$, which is used to compute the mean-square-error (MSE) with $\boldsymbol{z}^{(r)}$. The temporal layers of the U-Net are trainable to minimize the MSE loss.

For the left-to-left branch, the warped mask becomes a zero mask $\boldsymbol{M}_0$, indicating that there is no region for inpainting. Other steps remain identical to the left-to-right branch.

3.5. Inference

Consistent left and right view generation. During inference, the yellow modules in Figure 4 are omitted. Without ground truth depth, we use DepthCrafter [25] to obtain video depth $D^{(l)}$ and apply a fixed disparity scaling S of 0.03 for all test videos. We run both branches using the warped mask $\boldsymbol{M}$ as an indicator for left-to-right generation and $\boldsymbol{M}_0$ for left-to-left generation. After generating the latents $\hat{\boldsymbol{z}}^{(l)}$ and $\hat{\boldsymbol{z}}^{(r)}$ via iterative diffusion denoising, the VAE decoder is applied (as shown in the green boxes) to reconstruct the videos of both views $\hat{\boldsymbol{x}}^{(l)}$ and $\hat{\boldsymbol{x}}^{(r)}$.

Post-processing with histogram matching. Thanks to our handling of degraded inputs, our method produces stereo

Figure 3. Data augmentation with degradations is crucial for stereo-consistent generation when the input is degraded. With augmentation, our right-view output preserves sharper edge details compared to the version without augmentation. In contrast, StereoCrafter is not trained for degraded inputs, resulting in noisier and blurrier right-view generations. Input is degraded with Gaussian noise. The video of this example can be found in the Supplemental HTML.

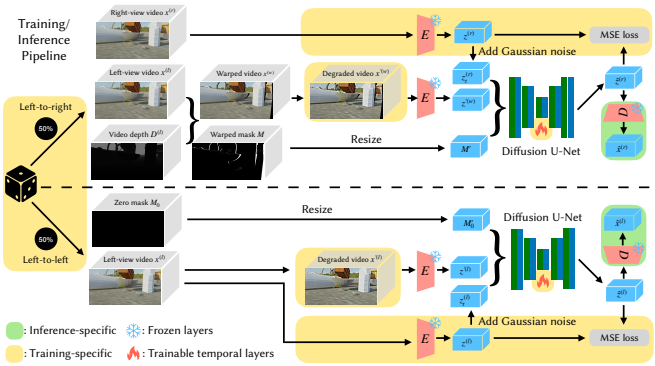

Figure 4. This figure shows our method's training and inference pipeline. We train/fine-tune the diffusion U-Net for two branches: left-to-right generation and restoration, and left-to-left restoration. During training, one of the branches is randomly selected. The left-to-right branch uses depth maps, forward warping, and the right-view target video. The left-to-left branch is nearly symmetric to the left-to-right branch, except that we use a zero mask as the condition to indicate that the task is restoration only, without any warping. Both branches incorporate data augmentation with degradations during training. During inference, we run both branches as well as the decoder to generate videos for both views, without using the yellow boxes. Note that the two U-Nets share the same weights and CLIP [24] features of $\boldsymbol{z}'^{(w)}$ and $\boldsymbol{z}'^{(l)}$ are also part of the conditional input to the U-Net, omitted for simplicity.

views that are structurally and visually consistent. However, the outputs may still exhibit slight differences in brightness or exposure relative to the input. To address this, we apply histogram matching using the input as the reference to post-process the right-view output, which better aligns brightness and exposure, as shown in Figure 5. Note that applying the same histogram matching to StereoCrafter does not improve its color consistency, indicating that our fine-tuned model has better photometric alignment and works better with post-processing.

4. Experiments

4.1. Implementation details

Data Generation. We use preprocessed ShapeNet [22] objects and environment maps [23] in Kubric to generate diverse training scenes. For each scene, we randomly sample stereo cameras with different baselines on spheres with different radii. We generated 958 stereo videos by moving objects forward, each containing 16 frames. For quantitative evaluation and user study, we collect 15 diverse monocular videos, including synthetic videos from Kubric [21], SVD [1], and real-world videos from CLEVR [27], Pixabay [6] and some self-captured videos. Besides, we evaluate our method on DAVIS [7] and Waymo [8] test sets. All test videos are not unseen during training.

Fine-tuning. We use the Stable Video Diffusion (SVD) architecture initialized with the trained weights of StereoCrafter. Following an open source implementation [28], we fine-tune the temporal transformer blocks and use the EDM sampler [29] for training and sampling. Optimization is performed using AdamW optimizer [30] with a learning rate of 2e-5 and a batch size of 1. The training takes 10,000 iterations in around 6 hours on an NVIDIA Tesla H100 GPU.

Inference time. Total inference time of different models is shown in Table 1 using an NVIDIA GeForce RTX 4090. Details of the timing breakdown of each model are shown in the Supplemental document.

4.2. Comparisons under various degradation types

In this section, we compare both qualitatively and quantitatively across different degradation types. Qualitatively, we provide zoom-in crops to better see the details. We also include video results in the Supplemental HTML for better viewing of the details. Quantitatively, we evaluate LPIPS [32] and CLIP [24] scores between corresponding left- and right-view frames (denoted as `view`), as well as temporal scores between consecutive right-view frames (denoted as `tempo`), following SVG [3]. We also evaluate

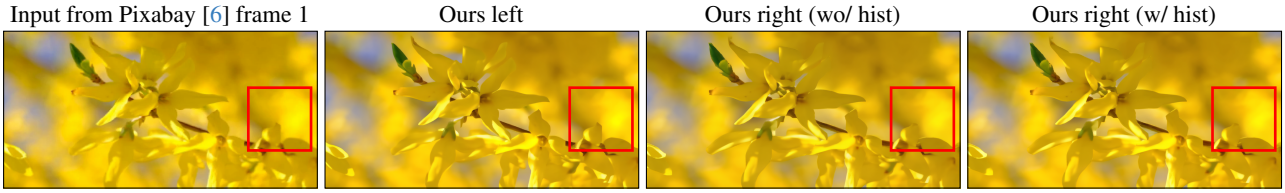

Figure 5. The color histogram of Ours with histogram matching between left and right is better matched than Ours without histogram matching. This can be better observed in the region around the red box, where the output becomes darker without histogram matching. Input is from Pixabay [6] being downsampling-degraded to 320 × 160.

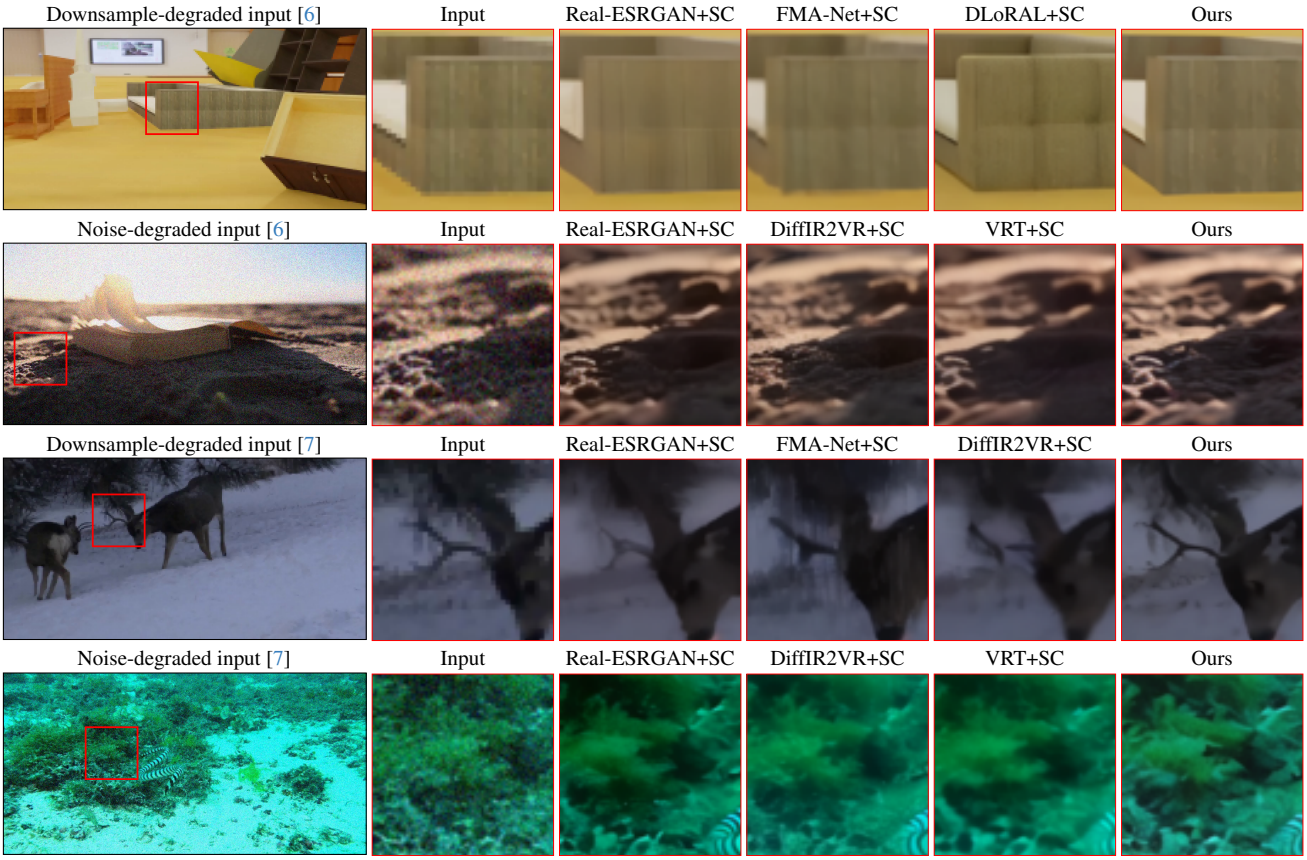

Figure 6. Comparisons on downsampled and noise-degraded input videos between two-stage methods (video restoration followed by StereoCrafter (SC) [9]) and our joint generation and restoration pipeline. Zoomed-in regions highlight that our method produces sharper and cleaner details across different scenes. Downsampled inputs are degraded to 320 × 160, 256 × 128 following Eq. (1), while noise-degraded inputs are corrupted with Gaussian noise of standard deviation 30. Corresponding quantitative and user study results are provided in Tables 1 and 4. Videos are provided in the Supplemental HTML for better visualization of temporal and view consistency.

restoration quality using PSNR, denoted as PSNR_{re} in Tables 1 and 2.

Downsampling. In the 1st and 3rd rows of Figure 6, the input videos are degraded by downsampling. Real-ESRGAN [13] produces sharper results but suffers from temporal inconsistency in wood textures and deer antlers, which the subsequent StereoCrafter [9] generation step cannot correct. FMA-Net [15] restores temporal consistency but fails to recover fine details. DLoRAL [14] upsamples textures with hallucinated, overly sharp details (1st row), making it difficult for StereoCrafter to generate stereo-consistent textures due to distribution mismatch. This issue is more obvious in the Supplemental HTML video. DiffIR2VR-Zero [16] yields grid artifacts. Our method achieves better temporal consistency and sharpness. As shown in Table 1, our joint approach also outperforms other methods in terms of LPIPS-

Table 1. Quantitative results on our 15-video test set. We show on average better view and temporal consistency, aligning with the user study in Table 4. We show the best, second-best, and third-best in the following. We use SC as the abbreviation of StereoCrafter in the table to save space.

Degradation	Method	$\text{LPIPS}_{\text{view}}$(↓)	$\text{LPIPS}_{\text{tempo}}$(↓)	$\text{CLIP}_{\text{view}}$(↑)	$\text{CLIP}_{\text{tempo}}$(↑)	PSNR_{re}(↑)	Time(s)(↓)
Downsampling	StereoCrafter [9] (abbrev. SC)	0.3083	0.1216	0.9124	0.9861	-	30
	StereoDiffusion [4]	0.3423	0.1987	0.9198	0.9718	-	160
	Real-ESRGAN [13]+SC	0.2522	0.1160	0.9226	0.9855	30.5088	35
	FMA-Net [15]+SC	0.2819	0.1222	0.9170	0.9857	30.6311	40
	DLoRAL [14]+SC	0.2781	0.1194	0.9135	0.9845	25.6845	62
	DiffIR2VR-Zero [16]+SC	0.2537	0.1502	0.9205	0.9830	28.9597	990
	Ours	0.2430	0.1228	0.9802	0.9867	31.0323	60
Noise addition	StereoCrafter [9]	0.5465	0.1797	0.8732	0.9763	-	30
	Real-ESRGAN [13]+SC	0.2709	0.1252	0.9179	0.9853	32.1156	40
	VRT [17]+SC	0.3052	0.1300	0.9174	0.9854	29.5914	125
	DiffIR2VR-Zero [16]+SC	0.2551	0.1299	0.9294	0.9864	31.4767	1230
	Ours	0.2551	0.1175	0.9796	0.9862	29.3928	60
No degradation	StereoCrafter [9]	0.2743	0.1217	0.9315	0.9857	-	30
	StereoDiffusion [4]	0.3495	0.1891	0.9332	0.9730	-	160
	SVG [3]	0.3574	0.1338	0.9567	0.9832	-	600
	Ours	0.2603	0.1264	0.9739	0.9865	-	60

Table 2. Quantitative results on the DAVIS [7] 30-video test set, comparing stereo view consistency and temporal consistency across methods. On average, our method achieves better view and temporal consistency while maintaining comparable performance to StereoCrafter when no degradation is applied. This suggests that the fine-tuned model remains generalizable to real-world videos from DAVIS [7]. Visual comparisons are partly shown in Figure 6 (last two rows). We mark the best, second-best, and third-best results. We use SC as the abbreviation of StereoCrafter in the table to save space.

Degradation	Method	$\text{LPIPS}_{\text{view}}$(↓)	$\text{LPIPS}_{\text{tempo}}$(↓)	$\text{CLIP}_{\text{view}}$(↑)	$\text{CLIP}_{\text{tempo}}$(↑)	PSNR_{re}(↑)	Time(s)(↓)
Downsampling	StereoCrafter [9] (abbrev. SC)	0.3922	0.2250	0.8919	0.9756	-	30
	Real-ESRGAN [13]+SC	0.3254	0.2073	0.8827	0.9743	27.4519	35
	FMA-Net [15]+SC	0.3960	0.2437	0.8844	0.9719	25.4420	40
	DiffIR2VR-Zero [16]+SC	0.3358	0.2046	0.8832	0.9771	27.1661	990
	Ours	0.3148	0.2220	0.9658	0.9762	27.7850	60
Noise addition	StereoCrafter [9]	0.5793	0.2536	0.8526	0.9687	-	30
	Real-ESRGAN [13]+SC	0.3169	0.2078	0.8914	0.9765	29.7016	40
	VRT [17]+SC	0.3282	0.2137	0.8851	0.9781	28.4164	125
	DiffIR2VR-Zero [16]+SC	0.3124	0.2172	0.8973	0.9769	28.6924	1230
	Ours	0.3162	0.2164	0.9601	0.9755	27.4447	60
No degradation	StereoCrafter [9]	0.3214	0.2235	0.9098	0.9777	-	30
	StereoDiffusion [4]	0.4234	0.3083	0.9168	0.9521	-	160
	SVG [3]	0.4103	0.2303	0.9505	0.9752	-	600
	Ours	0.3299	0.2261	0.9510	0.9771	-	60

and CLIP-based view and temporal scores. Table 2 shows that our method matches others under no degradation while improving performance on degraded data, indicating that fine-tuning preserves the base model capability while improving robustness to degradations.

User study. To evaluate human perception, we compare StereoCrafter [9], FMA-Net [15]+StereoCrafter, and our method. The generated 15 stereo videos of the 3 methods are shown to users using a VR headset (Meta Quest 3). We include FMA-Net as it exhibits fewer temporal inconsistencies than Real-ESRGAN [13]. Although Real-ESRGAN achieves slightly higher numerical scores, its artifacts are easily noticed in a VR headset. There are 15 computer science graduate students (3 female, 12 male) with normal or corrected vision participating in the user study. Each participant rated the methods on 3 criteria: 3D stereo effect, temporal consistency, and per-frame image quality, using a 1–5 scale. The results show that our method achieves higher scores in per-frame image quality, as well as improved stereo effect and temporal consistency.

Noise addition. Real-ESRGAN [13] suffers from temporal inconsistency similar to its behavior in upsampling. In Figure 6, DiffIR2VR-Zero [16] recovers details better but exhibits grid artifacts (2nd row). VRT [17] produces blurrier

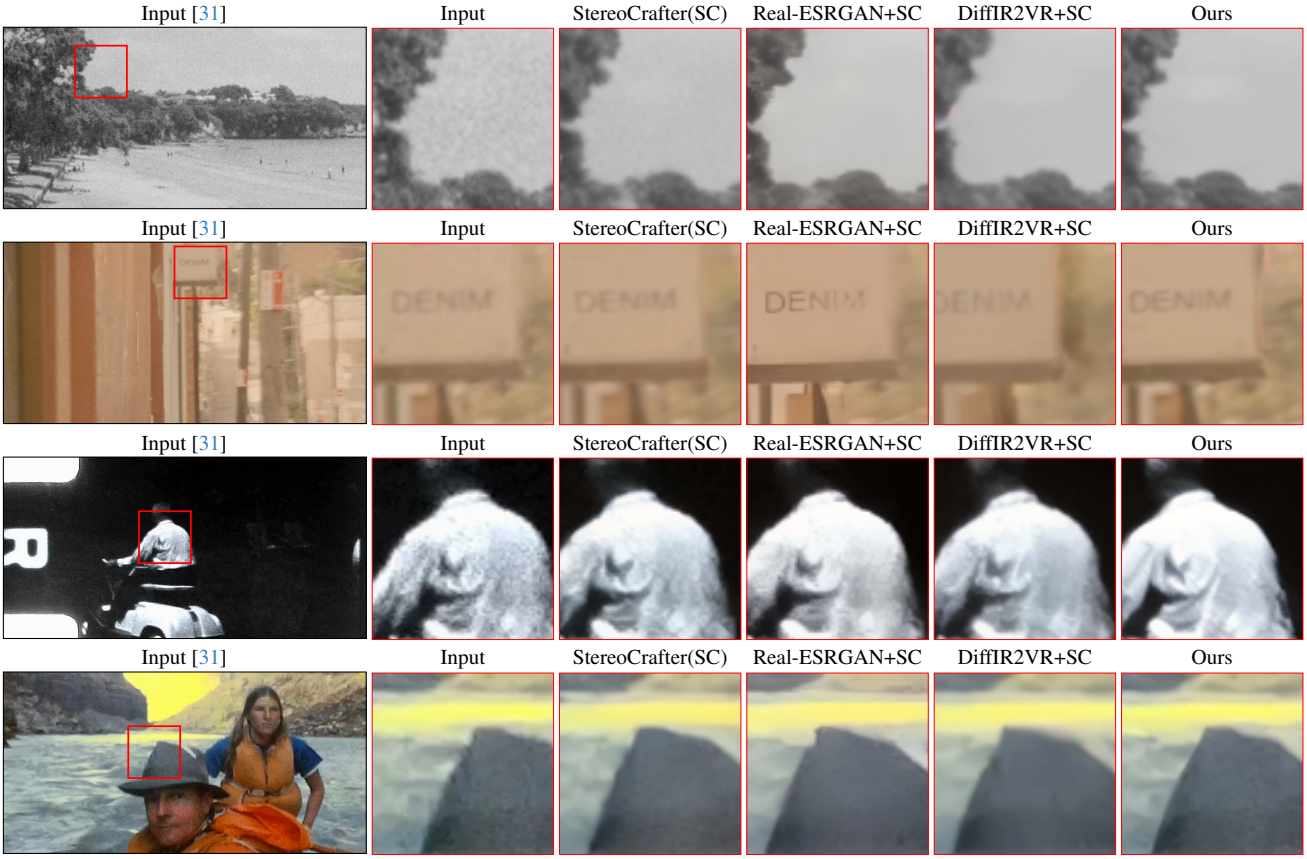

Figure 7. We compare methods on videos collected from Pexels [31] using the keywords "film grain noise and high ISO". As the noise comes from real camera capture, we only crop and resize the videos to 1024×512 without additional processing. We run four methods for comparison. Real-ESRGAN [13] restoration shows noticeable temporal flickering, and StereoCrafter alone does not remove the noisy artifacts. Our method achieves sharper details than DiffIR2VR [16] plus StereoCrafter. On average, our method achieves better stereo view consistency and temporal consistency. Quantitative results are shown in Table 5. The results again show that our fine-tuned model generalizes robustly to challenging real-world degraded videos.

Table 3. Comparison to GEN3C [19] on their released 4 long self-driving non-degraded videos from Waymo Open Dataset [8], in terms of view- and temporal-consistency.

Method	$\text{LPIPS}_{\text{view}}(\downarrow)$	$\text{LPIPS}_{\text{tempo}}(\downarrow)$	$\text{CLIP}_{\text{view}}(\uparrow)$	$\text{CLIP}_{\text{tempo}}(\uparrow)$
GEN3C	0.4864	0.1859	**0.9468**	0.9868
Ours	**0.4228**	**0.1808**	0.9366	**0.9901**

Table 4. User study scores on 3D Stereo Effect, Temporal Consistency across frames, and Image Quality per frame. We show both the average score of all users and scenes, as well as the standard deviation in brackets.

Metrics	StereoCrafter(SC)	FMA-Net+SC	Ours
3D Stereo Effect(↑)	3.81 (0.11)	3.68 (0.16)	**4.07** (0.10)
Temporal Consistency(↑)	3.96 (0.05)	3.81 (0.23)	**4.28** (0.12)
Image Quality(↑)	3.17 (0.22)	3.31 (0.19)	**4.48** (0.07)

Table 5. We evaluate on videos from Pexels [31], collected using the keywords "film grain noise and high ISO", with examples shown in Figure 7.

Method	$\text{LPIPS}_{\text{view}}(\downarrow)$	$\text{LPIPS}_{\text{tempo}}(\downarrow)$	$\text{CLIP}_{\text{view}}(\uparrow)$	$\text{CLIP}_{\text{tempo}}(\uparrow)$
StereoCrafter (abbrev. SC)	0.3716	0.2079	0.8877	0.9770
Real-ESRGAN+SC	0.3348	0.1920	0.8620	0.9747
DiffIR2VR-Zero+SC	**0.3200**	**0.1892**	0.8762	**0.9803**
Ours	0.3387	0.1969	**0.9254**	**0.9803**

results, though offering stable temporal consistency. While our method is not the strongest in denoising quality (PSNR in Tables 1 and 2), it achieves superior view and temporal consistency under varying degradation types, aligning with the goals of robust stereo video generation.

No degradation. StereoCrafter [9], SVG [3], and StereoDiffusion [4] are designed for inputs without degradation, so it is fair to compare them directly. Although we fine-tune

Table 6. Comparisons of training with different sizes of datasets on the quality of right-view video generation on the validation set of our synthetic data.

Method	LPIPS (↓)
Ours (fine-tuned wo/ augmentation)	0.3664
Ours (fine-tuned w/ 10 stereo videos)	0.3878
Ours (fine-tuned w/ 100 stereo videos)	0.3446
Ours (fine-tuned w/ 958 stereo videos)	**0.3422**

our model to handle more diverse inputs, its quality does not decrease in either visual or quantitative metrics (see Table 1). This confirms that our model remains strong on clean inputs while generalizing to varying types of degradations.

Tests on long videos. We further evaluate our method on long real-world autonomous driving videos. Using 4 released videos from GEN3C [19], we compare against their approach under the same Stable Video Diffusion backbone [1] (see Table 3). The videos are resized to 1024×512 without degradation and split into 16-frame segments for inference. Our method achieves results comparable to GEN3C, a multi-view video diffusion model.

Generalization to real-world videos. The results on DAVIS [7] in Figure 6 and Waymo [8] in Figure 1 already show that our fine-tuned model generalizes to real-world videos with and without degradations. Here we provide additional results on videos from Pexels [31], using "film grain noise and high ISO" as search keywords. Since the noise comes from real camera capture, we only crop and resize the videos to 1024×512 without further processing. Real-ESRGAN [13] and StereoCrafter [9] show noticeable temporal flickering, while StereoCrafter alone fails to remove the noisy artifacts. Our method produces more stable results with better stereo view and temporal consistency. Quantitative and user study results are shown in Table 5, with examples shown in Figure 7.

Robust to different disparity scaling. Our method is robust to various disparity scaling S. Even though we train on a fully black and warped right-view mask, we can control the output disparity by changing the disparity during warp mask generation. Visual results can be found in Supplemental Figure 3.

Takeaway. Across all degradation types, our method consistently achieves stronger view and temporal consistency than existing approaches, even without the highest restoration PSNR. This shows that our unified design is robust to diverse input qualities while maintaining high stereo consistency. In contrast, two-stage pipelines suffer from distribution mismatch, where artifacts from the first stage persist into the second (e.g., texture artifacts in Figure 6), and higher resource costs, requiring two separate models and either millions more parameters than ours or slower inference time.

4.3. Ablations

The importance of data augmentation with image degradations is shown in Figure 3. We further observe that the model performance improves with increasing training data, as shown in Table 6. With more diverse scenes (i.e., more training videos), our fine-tuned model achieves better right-view generation quality.

The effect on histogram matching is shown in Figure 5. This post-processing step further improves the color histogram alignment between the left and right views. The issue exists in the StereoCrafter base model and is more severe before applying our fine-tuning pipeline.

Regarding the architectural design, an alternative is a single-branch model that takes an input video and directly outputs two videos (i.e., a stereo pair). However, this leads to poor results, as it requires changing the output channels to generate two videos and retraining the final output layer from scratch.

5. Conclusion

We introduced a unified pipeline for joint stereo video generation and restoration. By using a consistent training and inference scheme conditioned on the warping mask, along with data augmentation, our method can be fine-tuned on a relatively small synthetic dataset and remains robust across varying levels of input degradation, unlike existing approaches that treat generation and restoration separately.

As shown in Table 1, our model is not the best on restoration alone, likely due to architectural limitations for this task. Future work may explore other backbones such as Diffusion Transformers (DiT) [33]. Our fine-tuning pipeline is architecture-agnostic and can potentially be transferred to a DiT-based stereo video generation model, although such a model is not yet available to our knowledge. This may further improve both restoration and generation quality, while moving toward a unified architecture supporting stereo generation, restoration, inpainting, and broader video processing as a step toward a video foundation model.

References

[1] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al., "Stable video diffusion: Scaling latent video diffusion models to large datasets," *arXiv preprint arXiv:2311.15127*, 2023. 1, 3, 4, 8

[2] Jian Shi, Qian Wang, Zhenyu Li, and Peter Wonka, "Stereocrafter-zero: Zero-shot stereo video generation with noisy restart," *arXiv preprint arXiv:2411.14295*, 2024. 1

[3] Peng Dai, Feitong Tan, Qiangeng Xu, David Futschik, Ruofei Du, Sean Fanello, XIAOJUAN QI, and Yinda Zhang, "SVG: 3d stereoscopic video generation via denoising frame matrix," in *The Thirteenth International Conference on Learning Representations*, 2025. 2, 4, 6, 7

[4] Lezhong Wang, Jeppe Revall Frisvad, Mark Bo Jensen, and Siavash Arjomand Bigdeli, "Stereodiffusion: Training-free stereo image generation using latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7416–7425. 2, 6, 7

[5] Qiao Jin, Xiaodong Chen, Wu Liu, Tao Mei, and Yongdong Zhang, "T-svg: Text-driven stereoscopic video generation," *arXiv preprint arXiv:2412.09323*, 2024. 1

[6] Pixabay, "Pixabay license summary," https://pixabay.com/service/license-summary/, 2025, Accessed: 2025-03-03. 1, 4, 5

[7] Anna Khoreva, Anna Rohrbach, and Bernt Schiele, "Video object segmentation with language referring expressions," in *Asian conference on computer vision*. Springer, 2018, pp. 123–141. 1, 4, 5, 6, 8

[8] 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*, 2020, pp. 2446–2454. 1, 4, 7, 8

[9] Sijie Zhao, Wenbo Hu, Xiaodong Cun, Yong Zhang, Xiaoyu Li, Zhe Kong, Xiangjun Gao, Muyao Niu, and Ying Shan, "Stereocrafter: Diffusion-based generation of long and high-fidelity stereoscopic 3d from monocular videos," *arXiv preprint arXiv:2409.07447*, 2024. 1, 2, 3, 5, 6, 7, 8

[10] Jiale Zhang, Qianxi Jia, Yang Liu, Wei Zhang, Wei Wei, and Xin Tian, "Spatialme: Stereo video conversion using depth-warping and blend-inpainting," *arXiv preprint arXiv:2412.11512*, 2024. 2

[11] Lukas Mehl, Andrés Bruhn, Markus Gross, and Christopher Schroers, "Stereo conversion with disparity-aware warping, compositing and inpainting," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 4260–4269.

[12] Jian Shi, Zhenyu Li, and Peter Wonka, "Immersepro: End-to-end stereo video synthesis via implicit disparity learning," *arXiv preprint arXiv:2410.00262*, 2024. 1

[13] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan, "Real-esrgan: Training real-world blind super-resolution with pure synthetic data," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 1905–1914. 2, 3, 5, 6, 7, 8

[14] Yujing Sun, Lingchen Sun, Shuaizheng Liu, Rongyuan Wu, Zhengqiang ZHANG, and Lei Zhang, "One-step diffusion for detail-rich and temporally consistent video super-resolution," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 2, 5, 6

[15] Geunhyuk Youk, Jihyong Oh, and Munchurl Kim, "Fma-net: Flow-guided dynamic filtering and iterative feature refinement with multi-attention for joint video super-resolution and deblurring," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 44–55. 2, 5, 6

[16] Chang-Han Yeh, Chin-Yang Lin, Zhixiang Wang, Chi-Wei Hsiao, Ting-Hsuan Chen, Hau-Shiang Shiu, and Yu-Lun Liu, "Diffir2vr-zero: Zero-shot video restoration with diffusion-based image restoration models," *arXiv preprint arXiv:2407.01519*, 2024. 2, 5, 6, 7

[17] Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool, "Vrt: A video restoration transformer," *IEEE Transactions on Image Processing*, 2024. 2, 6

[18] Aakash Garg, Libing Zeng, Andrii Tsarov, and Nima Khademi Kalantari, "Text2stereo: Repurposing stable diffusion for stereo generation with consistency rewards," *arXiv preprint arXiv:2506.05367*, 2025. 2

[19] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao, "Gen3c: 3d-informed world-consistent video generation with precise camera control," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 6121–6132. 2, 7, 8

[20] Shangchen Zhou, Peiqing Yang, Jianyi Wang, Yihang Luo, and Chen Change Loy, "Upscale-a-video: Temporal-consistent diffusion model for real-world video super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 2535–2545. 2

[21] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al., "Kubric: A scalable dataset generator," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 3749–3761. 3, 4

[22] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al.,

"Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015. 3, 4

[23] Yannick Hold-Geoffroy, Akshaya Athawale, and Jean-François Lalonde, "Deep sky modeling for single image outdoor lighting estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6927–6935. 3, 4

[24] 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*. PmLR, 2021, pp. 8748–8763. 4

[25] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan, "Depthcrafter: Generating consistent long depth sequences for open-world videos," *arXiv preprint arXiv:2409.02095*, 2024. 3

[26] Alexey Bochkovskiy, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan Richter, and Vladlen Koltun, "Depth pro: Sharp monocular metric depth in less than a second," in *The Thirteenth International Conference on Learning Representations*, 2025. 3

[27] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2901–2910. 4

[28] Pengxiang Li, Zhili Liu, Kai Chen, Lanqing Hong, Yunzhi Zhuge, Dit-Yan Yeung, Huchuan Lu, and Xu Jia, "Trackdiffusion: Multi-object tracking data generation via diffusion models," *arXiv preprint arXiv:2312.00651*, 2023. 4

[29] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine, "Elucidating the design space of diffusion-based generative models," *Advances in neural information processing systems*, vol. 35, pp. 26565–26577, 2022. 4

[30] I Loshchilov, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017. 4

[31] Pexels, "Pexels license," 2026, Accessed: 2026-03-07. 7, 8

[32] 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*, 2018, pp. 586–595. 4

[33] William Peebles and Saining Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4195–4205. 8