

# M2SVid: End-to-End Inpainting and Refinement for Monocular-to-Stereo Video Conversion. Supplementary Material

Nina Shvetsova<sup>1,2,3,4\*</sup>, Goutam Bhat<sup>1</sup>, Prune Truong<sup>1</sup>, Hilde Kuehne<sup>2,3</sup>, Federico Tombari<sup>1,5</sup>

<sup>1</sup>Google, <sup>2</sup>Tuebingen AI Center/University of Tuebingen, <sup>3</sup>Goethe University Frankfurt,

<sup>4</sup>MPI for Informatics, Saarland Informatics Campus, <sup>5</sup>Technical University of Munich

Project webpage: <https://m2svid.github.io/>

The supplementary material provides more details about the datasets, implementation, and results. Appendix A provides a brief background on Diffusion Models. More details about our training and evaluation datasets are provided in Appendix B. Appendix C provides further implementation details as well as additional information on user studies, while Appendix D contains more detailed results and analysis. Appendix E provides details on inference for high-resolution long videos. Finally, Appendix F discusses the limitations of our approach. We also include the results generated by our methods as well as the compared approaches, on the 21 videos used for our user study. You can view each of them independently, or open the *index.html* file in a browser. We also provide additional results generated by our model on high-resolution long videos as described in Appendix E.

## A. Background on Diffusion Models

**DDPMs:** Denoising Diffusion Probabilistic Models (DDPMs) [6] are generative models trained to map a simple noise distribution  $p_T$  to the data distribution  $p_0$ , by reversing a stochastic forward process  $p_t$ ,  $t = 1, \dots, T$ . This forward process gradually adds small amounts of Gaussian noise with variance  $\beta_t$ , ensuring the reverse process can be approximated as a Gaussian distribution. The forward process might be also defined by  $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$ , where  $\mathbf{x}_0$  is a data sample,  $\epsilon \sim \mathcal{N}(0, \mathbf{I})$  is a noise, and  $\alpha_t = 1 - \beta_t$  and  $\bar{\alpha}_t = \prod_{\tau=1}^t \alpha_\tau$ . A denoising model  $\hat{\mathbf{v}}_\theta(x_t, t)$  is trained to progressively remove noise from the input, by predicting  $\mathbf{x}_{t-1}$  from  $\mathbf{x}_t$  and timestamp  $t$ , effectively reversing the diffusion.

**DDIMs:** Denoising Diffusion Implicit Models (DDIMs) [15] extend DDPMs by introducing non-Markovian diffusion processes, preserving the original training objective while enabling much faster inference with as few as 25 or 50 steps.

**LDMs:** To facilitate training, Latent Diffusion Models (LDMs) [12] operate in the latent space of a Variational Autoencoder (VAE) [8], which consists of an encoder  $E$  and a decoder  $D$ . The VAE is trained separately and remains frozen during diffusion model training. By working in the latent space, where  $D(E(x)) \approx x$ , LDMs significantly reduce computational complexity while preserving essential data structure. The diffusion model  $\hat{\mathbf{v}}_\theta$  is then trained directly in this lower-dimensional latent space.

**Conditional diffusion models:** Finally, conditional diffusion models [13, 17] extend diffusion models by conditioning the denoising model  $\hat{\mathbf{v}}_\theta(x_t, t, c)$  on additional input  $c$ , such as text [13], images [13], or pose and depth maps [17].

## B. More Details on Datasets

### B.1. Stereo4D

**Stereo4D:** The dataset contains  $\sim 200\text{K}$  video clips sourced from  $\sim 7.5\text{K}$  online videos. Each clip lasts between 2–6 seconds with a spatial resolution of  $512 \times 512$ . The videos cover diverse scenes featuring both indoor and outdoor activities. While the dataset is relatively large, it also includes many static videos or videos with low disparity. We hold out a random subset of 400 videos for method evaluation and use the rest for the training. We source only one clip per video that is longer than 3.2 seconds and sample 16 frames at 5 fps, resulting in a test set of 400 diverse video clips with a resolution of  $16 \times 512 \times 512$ .

### B.2. Ego4D

**Ego4D:** This is one of the largest egocentric video datasets, where participants were asked to perform various activities. While most videos are monocular, the dataset includes 263 long stereoscopic videos (80 hours in total) with a resolution of  $1400 \times 1400$ . As the videos are egocentric, captured using a head-mounted camera, they exhibit significant camera movement and contain close objects with a large disparity range between the stereo views. We split 263 stereo videos

\*This work was conducted during an internship at Google.

kite-walk, Please assign a rank to each method, 1 being the best, 4 being the worst.



Figure 1. The interface shown to the users during the human perception study. On the left, we showed the input left video, together with the disocclusion mask. For each video, we showed results of the four method. Note that the users were not shown any labels to indicate which outputs were generated by which method. Furthermore, we randomized the order in which the methods were listed for each video. The users were asked to assign rankings from 1 (best) to 4 (worst) for each method. In cases two or more methods were indistinguishable, the users were given the choice of assigning the same rank to multiple methods. The users could play or pause the videos as they wish, view the video in full screen, or change the playback speed if they desired.

### Video Quality User Study

Given a monocular video as input, we aim to render the same video from a virtual right camera to enable more immersive viewing. The right view is generated by first reprojecting the left view to the right camera using estimated monocular depth. Next, we use different methods to fix the reprojection artifacts and inpaint the holes in the reprojected image. Below we show the inpainting results from 4 different methods, all of which use the same monocular depth for reprojection. The goal of this study is to rank the inpainting methods based on overall video quality.

**Instructions:** For each video, we show results from 4 different methods. In the left, for each video, we also show the inpainting mask on top (white = inpainted, black = not inpainted), and the input left video in the bottom. Please rank the methods based on overall video quality. Factors to consider include temporal consistency, image quality (i.e. sharpness), lack of artifacts, etc. **The ranking should be from 1 to 4, with 1 being the best method, and 4 being the worst method.**

The users are encouraged to pause the videos and comparing a few individual frames to help with the ranking. You may also view the video in full screen, or change the playback speed.

Next to each video, we show 4 buttons which are used to indicate rankings for each method. For example, the buttons at top-left corner correspond to the method shown in top-left of the video. Please click on the corresponding button to indicate your ranking for the method. **In case two or more methods are indistinguishable, please assign the same rank to them.**

**Submission:** After ranking all videos, click the "Submit All Rankings" button in the bottom. If all the rankings were filled, a CSV file containing the ratings will be downloaded.

Figure 2. The users were shown the above intructions during the study.

of the dataset into  $\sim 57k$  5-second video clips. We hold out all clips from one video, namely 200 clips, for method evaluation, while using the rest 57K clips for training.

**Ego4d Pre-processing:** We perform rectification for each long video as described in Sec. 5.5 of the main paper. For video rectification, we uniformly sample 200 frames from each video. After rectifying the videos, we spatially crop the video to discard boundary regions with missing pixels. Then, we split the videos into 5-second subsequent video clips and compute shifts and disparity at the clip level. Finally, to ensure high-quality rectified data, we perform LoFTR feature matching [16] again and filter point pairs by computing the fundamental matrix with RANSAC [4], applied to frames sampled at 1 FPS from each clip. We then filter out clips where the matched points exhibit more than 2 pixels of vertical disparity.

### B.3. DAVIS

**DAVIS:** The dataset consists of 50 videos, each up to 6 seconds long, with a resolution of  $1024 \times 1920$ . The videos are highly dynamic, featuring moving cameras and one or more moving objects (e.g., BMX riding, breakdancing, etc.). Note that the dataset only contains monocular videos, without a ground truth depth or ground truth right view. Hence we use this dataset solely for qualitative comparisons and user studies.

## C. Implementation Details

### C.1. Conditioning

We extend the first convolution of SVD that initially took 8-dimensional input (4 dim noise latent + 4 dim VAE encoded image) to the 13-dimensional input (4 dim noise latent + 4 dim VAE encoded left to view video + 4 dim VAE encoded warped view video + 1 dim disocclusion mask). To ensure smooth model initialization we copy weights 5-8 channels

Method	Training data	Denoising steps	Full view			Inside disocclusion mask only			Outside disocclusion mask		
			PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓
SVG [3]	~ (training free)	50 steps	25.6	0.926	0.217	38.1	0.994	0.014	26.3	0.940	0.190
StereoCrafter [14]	private dataset	25 steps	24.9	0.909	0.242	38.2	0.995	0.012	25.5	0.922	0.217
StereoCrafter [14]	private dataset	1 step	25.3	0.911	0.262	38.9	0.996	0.014	25.8	0.923	0.234
M2SVid (Ours)	Stereo4D + Ego4D	1 step	26.2	0.915	0.180	38.2	0.994	0.010	26.8	0.924	0.161

Table 1. Quantitative comparison of our approach with state-of-the-art methods on Stereo4D test set, in terms of PSNR, MS-SSIM, and LPIPS. We report the metrics computed over the full image, only inside the disocclusion mask, and only outside the disocclusion mask.

Model Architecture	End-to-end training	Denoising steps	Full image						Inside disoccluded regions					
			Stereo4D			Ego4D			Stereo4D			Ego4D		
			PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓
Cond. with $V_r^{warp} + M_r^{occ}$	✗	25 steps	24.5	0.886	0.226	20.8	0.822	0.296	36.1	0.993	0.011	29.3	0.986	0.027
Cond. with $V_r^{warp} + M_r^{occ} + V_l$	✗	25 steps	24.8	0.891	0.215	22.1	0.870	0.267	36.7	0.994	0.011	30.1	0.988	0.024
Cond. with $V_r^{warp} + M_r^{occ}$	✓	1 step	26.1	0.913	0.187	21.5	0.837	0.276	38.0	0.994	0.010	30.4	0.987	0.026
Cond. with $V_r^{warp} + M_r^{occ} + V_l$	✓	1 step	26.2	0.915	0.179	22.8	0.886	0.244	38.2	0.995	0.009	30.9	0.989	0.024
+full attention	✓	1 step	26.2	0.915	0.180	22.7	0.885	0.248	38.2	0.994	0.010	30.7	0.989	0.025

Table 2. We analyse the impact of different conditioning inputs as well as inference modes on Stereo4D and Ego4D datasets. The metrics are reported over the full images, as well as only the disoccluded regions.

to 9-12 channels and divide both by factor 2 and initialize channels of dissolution mask with 0.

**Model training:** We initialize our M2SVid model using the Stable Video Diffusion model (stable-video-diffusion-img2vid-xt version) [1]. We train the model for 300k iterations with a batch size of 16 and a learning rate of  $2 \times 10^{-6}$ . Due to GPU memory limitations, during training, we sample batches with frame number and resolutions of  $4 \times 512 \times 512$ ,  $16 \times 256 \times 256$ , and  $25 \times 192 \times 192$ , utilize gradient checkpointing[2] and train the model in float16 precision. We employ random resized cropping, using a resize scale in the range of [0.3, 1.0], for data augmentation. We preprocess the disocclusion masks using the closing morphological transformation to fill small holes inside the inpainting regions with a kernel size of 11 during training and testing.

**Depth estimation with DepthCrafter:** During inference for depth prediction, we utilize the DepthCrafter [7] method, the state-of-the-art in diffusion-based video-depth estimation approach. This model can accommodate up to 110 frames and output temporally consistent depth. However, any depth model might be used at this step, for example, efficient 1-step depth diffusion models as in [5]. We predict depth in chunks of 110 frames with an overlap of 25 frames. Depth was predicted on frames resized to a resolution of 1024, followed by upscaling to the original size.

## C.2. Desktop User Study

In order to perform our desktop human perception user study to rank the different methods, we selected a set of 21 videos. We tried to ensure a diverse collection of videos covering different types of subjects, video quality, and motion pattern. We avoided including without with very large

motions since we found it very hard to compare the methods in such cases.

During the study, the participants were provided HTML pages containing the links to the generated predictions, as shown in Figure 1. Furthermore, the users were shown detailed instructions about the study at the top of the HTML page. The exact instructions are shown in Figure 2. In particular, the users were provided a brief background about the problem statement and the method. They were asked to rank the quality of the different methods based on temporal consistency, image quality (i.e. sharpness), lack of artifacts, *etc.* In order to minimize noise, the users were allowed to assign same ranks to two or more methods in case they were indistinguishable.

We had 13 distinct participants in the study, out of which 9 were male and 4 were female. We divided the 21 videos into sets of 7 each. Each user was given the option of assigning rankings to the videos in one or two sets each. In total, we received rankings for 16 sets, resulting in 112 sets of ranks.

## C.3. VR Headset User Study

For the VR headset user study, the participants were shown the stereoscopic videos generated by our method and StereoCrafter (25 denoising steps) on a video player that supports side-by-side (3D) content. We used the same 21 videos that were used in the Desktop user study. The participants were then asked to rank the methods based on general viewing experience, while allowing for tied ratings. Note that all the videos were anonymized and the order of showing the methods was also randomized. We had 5 distinct participants in the user study (3 male and 2 female), each of whom provided rankings for all 21 videos.

Loss training	End-to-end Inference steps	Full image									Inside disoccluded regions					
		Stereo4D			Ego4D			Stereo4D			Ego4D					
		PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓
A Standard diffusion loss	✗	25 steps	24.8	0.891	0.215	22.1	0.870	0.267	36.7	0.994	0.011	30.1	0.988	0.024		
B Standard diffusion loss	✗	1 step	25.7	0.901	0.242	23.0	0.877	0.320	38.5	0.995	0.012	31.5	0.990	0.030		
C $L_{latent}$	✓	1 step	25.6	0.901	0.238	22.9	0.875	0.318	38.3	0.995	0.012	31.4	0.990	0.030		
D $L_{latent} + L_{LPIPS} + L_{L1}$	✓	1 step	26.2	0.915	0.179	22.8	0.886	0.244	38.2	0.995	0.009	30.9	0.989	0.024		

Table 3. We analyse the impact of different training strategies and losses on the Stereo4D and Ego4D datasets. Metrics are reports over the full image, as well as over only the disoccluded regions.

## D. More Details on Results

### D.1. State-of-the-art Methods

**SVG** is a training-free method that utilizes a frozen diffusion model to inpaint regions within the mask while preserving the other regions. To achieve spatial-temporal consistency, the method employs inpainting of 8 uniformly sampled views between the given left view and the required right view at the same time, increasing the computational burden. We use 50 denoising steps as recommended. SVG [3] has been compared to state-of-the-art video inpainting [9, 19] and dynamic novel view synthesis [10, 11] methods, significantly outperforming them in temporal consistency and overall inpainting quality. This highlights the fundamental differences between stereo video inpainting, classical video inpainting, and novel view synthesis, revealing a substantial gap between these tasks. Due to this disparity, we exclude classical video inpainting and novel view synthesis methods from our comparisons.

**StereoCrafter** [14] is a concurrent method and the closest baseline to our approach. It fine-tunes a diffusion model, SVD, to perform inpainting given a warped view and a mask in a diffusion-based denoising setup, trained on a privately collected internet dataset. Since the method does not specify the number of denoising steps, we use 25 steps, as is commonly done [7].

Note that our method M2SVid was trained on the Stereo4D and Ego4d datasets, while SVG is a training-free method, and StereoCrafter is trained on a private dataset. However, this StereoCrafter private dataset also contains general internet videos and should have a similar data distribution. Comparing zero-shot performance on another dataset is challenging, as there are no publicly available stereo video datasets in the general domain. For this reason, we evaluate on the Stereo4D dataset.

### D.2. State-of-the-art Comparison

For quantitative evaluation, we use PSNR, MS-SSIM, and LPIPS [18] metrics, computed independently for each video and averaged over the full dataset. Additionally, we report these metrics separately for the disoccluded and non-disoccluded regions in Tab. 1. To achieve this, we mask all pixels outside the considered region with white pixels, compute the metrics at the video level, and then average them

over the dataset.

Our method obtains the best PSNR and LPIPS scores when averaged over the full image. For the disoccluded regions, we see that single step StereoCrafter obtains the best PSNR score, while also having the worse LPIPs score. We believe this is because the method generates blurry results, which is often favored by PSNR, but strongly penalized by LPIPS. Outside the disocclusion region, our method obtains the best LPIPS and PSNR scores while SVG obtains the best MS-SSIM score.

### D.3. Full-attention at Disoccluded Pixels.

In Figs. 4 to 6, we provide further qualitative evaluation of our proposed full attention at disoccluded pixels. We find that full attention is particularly beneficial for scenes with complex backgrounds and strong camera movement, where both foreground and background pixels move and correct inpainting requires the model to “copy” disoccluded pixels from different spatial locations in other frames. For example, in Fig. 4, the handrail of the stairs in frame 4 can only be correctly inpainted by using information from a different spatial location in frame 9 due to camera movement. In Fig. 5, due to the strong movement of the motorcyclist behind the central rider, to inpaint the region near the helmet in frame 11, the model has to use data from a strongly shifted spatial location in frame 10. Finally, in Fig. 6, while standard factorized attention hallucinated the reins in frame 16 at the same spatial location as in the previous frame, creating a visual artifact as if the reins split into two pieces, full attention at disoccluded pixels prevents this hallucination and correctly inpaints the region.

### D.4. Ablations

In Tabs. 2 and 3, we further report metrics for the disoccluded regions for our ablation studies. We find that incorporating the left view as an additional conditioning signal (Tab. 2) results in a slight improvement in inpainting performance across both datasets. Additionally, end-to-end training with latent space supervision (Tab. 3) achieves nearly the same inpainting performance as a model trained with the standard diffusion paradigm and evaluated with 1-step inference. However, end-to-end training with image-based loss leads to a substantial drop in LPIPS, particularly on the Ego4D dataset.



Method	Latency (s) ↓
without full attention on the disoccluded regions	<b>2.0</b>
with full attention on the disoccluded regions	<b>2.1</b>

Table 4. The latency of two variations of our model: with and without full attention on the disoccluded regions. Enabling full attention on the disoccluded regions results in only a slight increase in runtime. The run-times computed using an A100 GPU on a  $512 \times 512$  videos with 16 frames.

Training regime	Loss	End-to-end training	Inference steps	LPIPS↓	
				Stereo4D	Ego4d
Sampling $t$	$L_{latent}$	✗	1 ( $t = T$ )	0.242	0.320
			5	0.217	0.278
			25	0.215	0.267
Fixed $t = T$	$L_{latent} + L_{LPIPS} + L_{L1}$	✓	1 ( $t = T$ )	<b>0.179</b>	<b>0.244</b>
			5	0.200	0.260
			25	0.243	0.298

Table 5. The effect of the number of inference steps on model performance. In standard training with sampled  $t$ , more inference steps improve results, while 1-step inference performs poorly. In contrast, our model is trained with fixed  $t = T$ , enabling end-to-end supervision in image space (which is not possible with sampled  $t$ ). This leads to superior 1-step performance, while multi-step inference degrades results due to mismatch with training.

## D.5. Run-time

We compare the runtime of our model with and without full attention in the disoccluded regions in Tab. 4. Our results show that enabling full attention leads to only a slight increase in runtime.

## D.6. Number of Denoising Steps

In Tab. 5, we analyze the effect of the number of inference steps on model performance. In a standard training regime, where a timestamp  $t$  is randomly sampled during training, one-step inference results in poor performance, while increasing the number of inference steps improves it. In contrast, we explicitly train a one-step model by fixing the timestamp to  $t = T$  during training. This enables end-to-end training with an image-space loss, as the U-Net directly predicts denoised latents that can be decoded with a VAE and supervised in image space. Note that training with an image-space loss is not possible in the standard regime with sampled  $t$ , since when  $t \neq T$ , the U-Net outputs a mixture of latents and noise (not clean latents), which therefore cannot be decoded with the VAE and supervised in image space. As a result of end-to-end training with a single step, our model achieves the best performance with one-step inference. However, multi-step inference degrades performance, as the model was not trained with  $t \neq T$ .

## E. Inference on Arbitrary Length and Resolution

To achieve stereo video conversion for videos of arbitrary length, we fine-tune our M2SVid model for 20K iterations using the auto-regressive modeling approach proposed in [14]. Specifically, during training, we replace the first  $n$  frames of  $V_r^{warp}$  with ground truth frames from the right video  $V_r$ , where  $n$  is randomly sampled between 0 and  $N$ . During inference on long videos, we use the last  $m$  frames generated in the current round as input for the next round, ensuring seamless stitching between consecutive rounds. For inference with arbitrary resolution, following [14], we utilize tiled diffusion. Specifically, we divide high-resolution videos into overlapping blocks and perform stereo video conversion on each block independently. The overlapping regions are blended in the latent space before VAE decoding to ensure seamless transitions between blocks. We provide examples generated by our model on high-resolution long videos in the supplementary material zip file.

## F. Limitations

We build our model upon the Stable Video Diffusion model, which allows us to leverage learned video priors obtained through large-scale generative pretraining. Due to the high computational cost of processing video data, SVD employs a VAE to compress videos into a lower-resolution latent space, where denoising is performed. Following this architecture, we first encode the video using a VAE, then apply a U-Net, and finally decode the output back to video space using the same VAE. However, VAE compression may lead to loss of high-frequency details. As illustrated in Fig. 3, even encoding and decoding the left-view videos with the VAE alone can introduce visible blurriness (e.g., the shoe in Figs. 3a and 3b) and loss of fine details. To support stereo conversion for higher-resolution videos, we additionally apply temporal and spatial stitching, as described in Appendix E. While simple and efficient, stitching may introduce further blurring artifacts, as shown in Figs. 3c and 3d. We plan to address these limitations in future research.

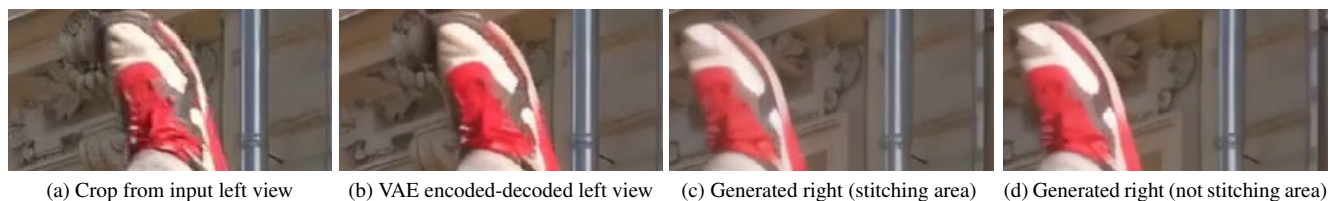


Figure 3. Blurriness due to usage of VAE and spatial stitching in high-resolution videos.

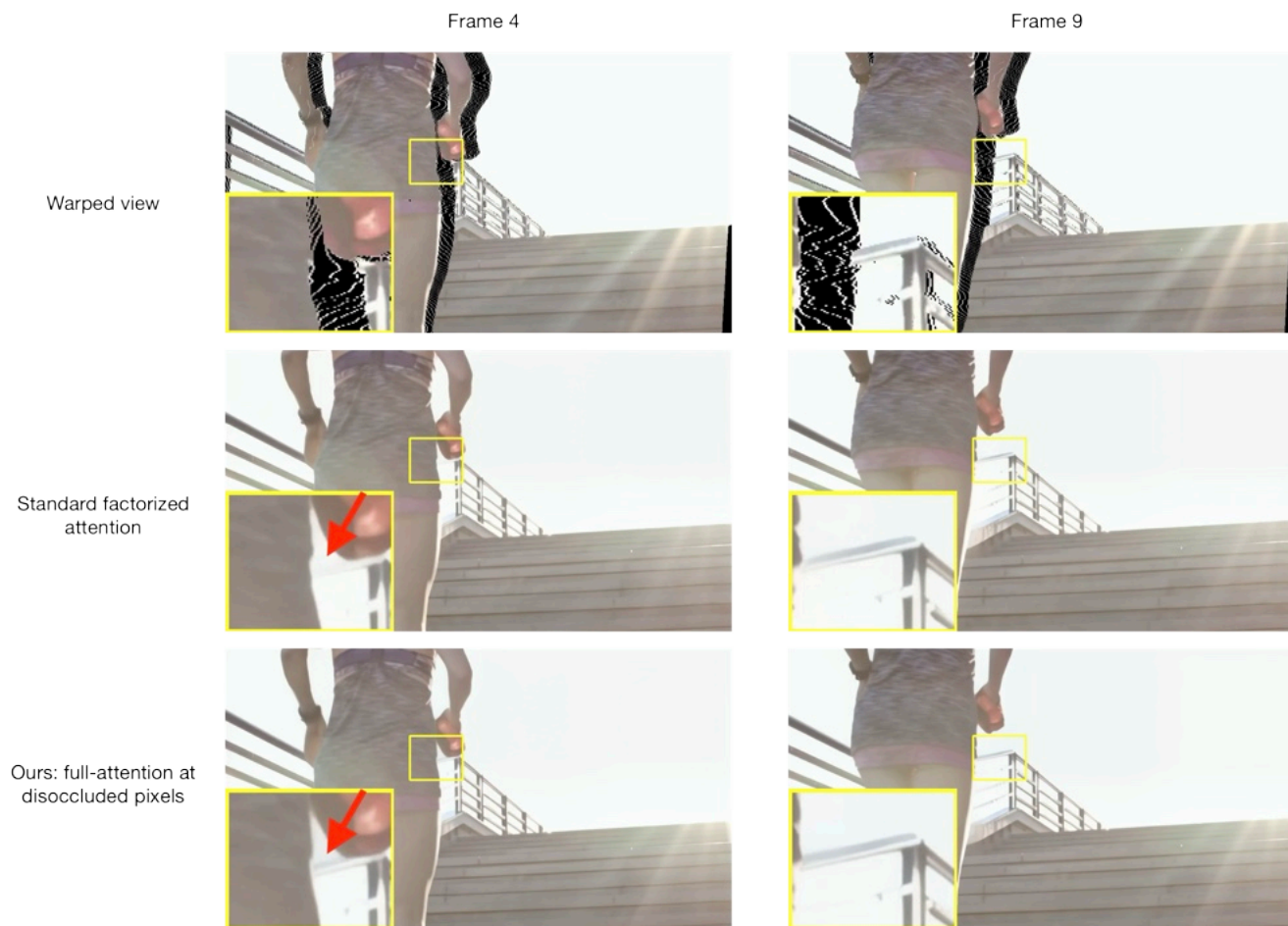


Figure 4. Standard factorized attention vs full-attention at dis-occluded pixels (ours). Full attention at dis-occluded pixels helps the model better exploit information from other frames, in particular by correctly inpainting the handrail of the stairs using data from a different spatial location in another frame.

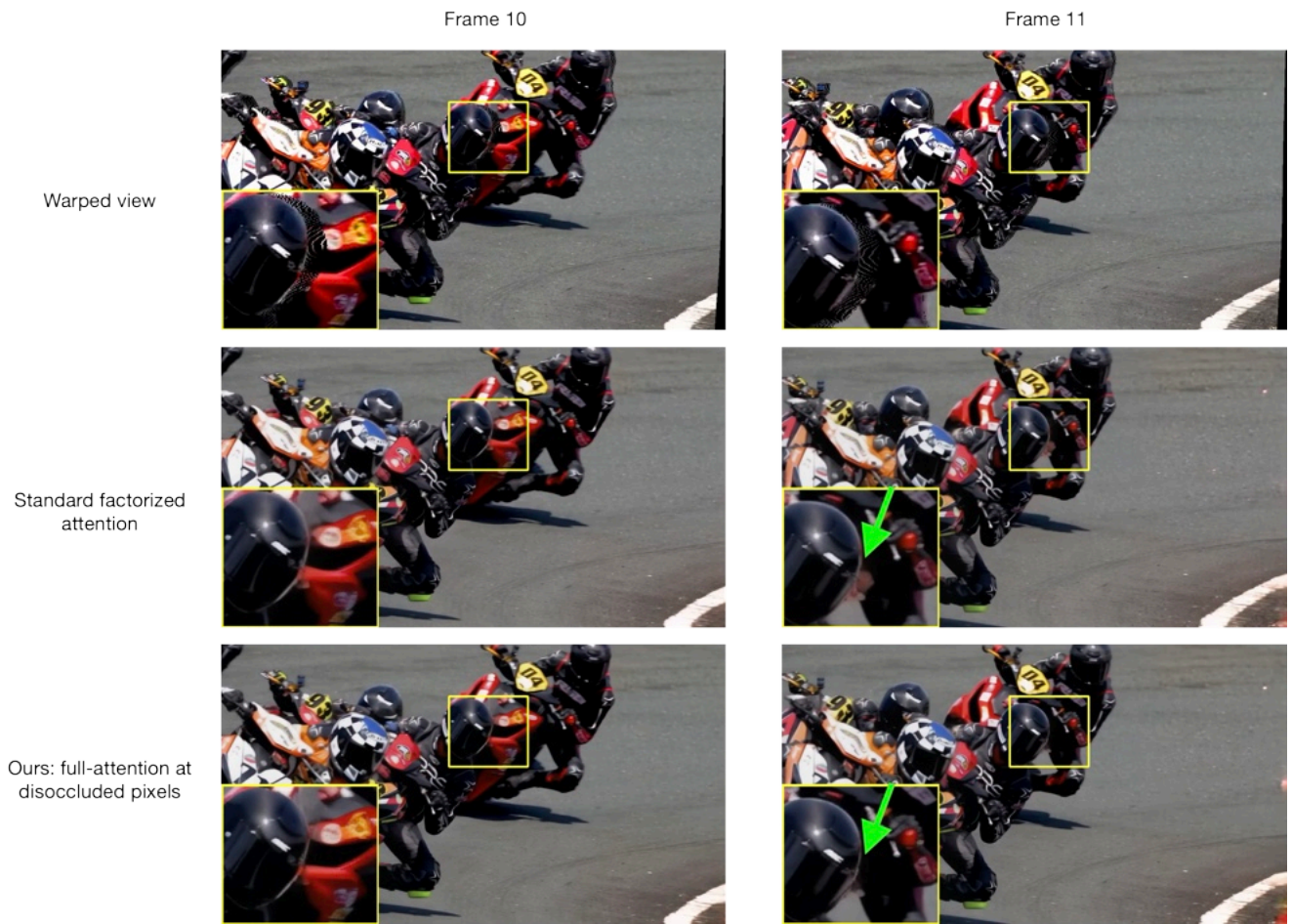


Figure 5. Standard factorized attention vs full-attention at dis-occluded pixels (ours). Full attention at dis-occluded pixels helps the model better exploit information from other frames, in particular by correctly inpainting the region near the helmet, using data from a different spatial location in another frame.



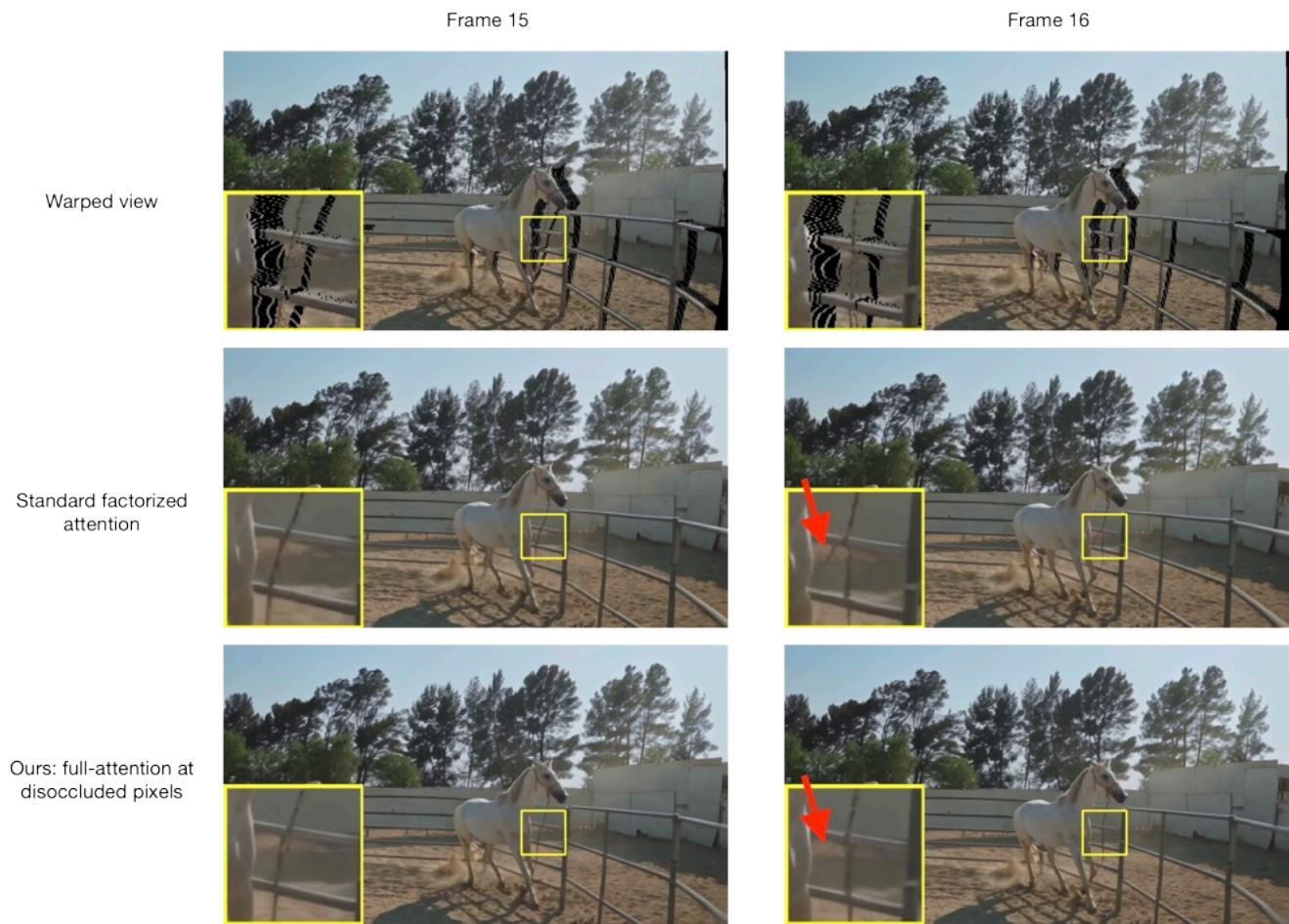


Figure 6. Standard factorized attention vs full-attention at dis-occluded pixels (ours). Full attention at dis-occluded pixels helps the model better exploit information from other frames. In particular, while standard factorized attention hallucinated the reins at the same spatial location as in the previous frame, creating a visual artifact as if the reins split into two pieces, full attention at dis-occluded pixels prevents this hallucination and correctly inpaints the region.



## 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. 3
- [2] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016. 3
- [3] Peng Dai, Feitong Tan, Qiangeng Xu, David Futschik, Ruofei Du, Sean Fanello, Xiaojuan Qi, and Yinda Zhang. Svc: 3d stereoscopic video generation via denoising frame matrix. *arXiv preprint arXiv:2407.00367*, 2024. 3, 4
- [4] MA FISCHLER AND. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 2
- [5] Gonzalo Martin Garcia, Karim Abou Zeid, Christian Schmidt, Daan de Geus, Alexander Hermans, and Bastian Leibe. Fine-tuning image-conditional diffusion models is easier than you think. *arXiv preprint arXiv:2409.11355*, 2024. 3
- [6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 1
- [7] 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, 4
- [8] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013. 1
- [9] Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. Towards an end-to-end framework for flow-guided video inpainting. In *CVPR*, 2022. 4
- [10] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *CVPR*, 2023. 4
- [11] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *CVPR*, 2023. 4
- [12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1
- [13] 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. *NeurIPS*, 2022. 1
- [14] 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. 3, 4, 5
- [15] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1
- [16] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loft: Detector-free local feature matching with transformers. In *CVPR*, 2021. 2
- [17] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 1
- [18] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 4
- [19] Shangchen Zhou, Chongyi Li, Kelvin CK Chan, and Chen Change Loy. Propainter: Improving propagation and transformer for video inpainting. In *ICCV*, pages 10477–10486, 2023. 4