TLB-VFI: Temporal-Aware Latent Brownian Bridge Diffusion for Video Frame Interpolation

Supplementary Material

1. Overview

The supplementary material is structured as follows:

- Sec. 2 includes optical basics: what are optical flows and warping.
- Sec. 3.1 contains PSNR/SSIM evaluated on our selected datasets.
- Sec. 3.2 contains additional qualitative results.
- Sec. 4.1 contains implementation details.
- Sec. 4.2 contains additional details on 3D wavelet transforms.
- Sec. 4.3 contains the proof to our proposition in Sec. 3.3 of our main paper.

2. Optical Flow Basics

Optical flow is the pixel-wise movement from frame to frame. If we have two images I_0 and I_1, and for a given pixel $I_0[i, j]$ the corresponding pixel appears in I_1 $I_1[i', j']$ then $flow(I_0, I_1)[i, j]$ is $[i' - i, j' - j]$, indicating the pixel movement. Warping is to move each pixel according to the movements defined by optical flows. Importantly, optical flows do not explicitly estimate the motion speed, and motion speed is implicitly contained in training data. If all training data consists of constant speed motion, and the test data contains motion speed that is not evenly distributed between two frames, the predicted position will not align. This would be an interesting research problem but out of scope of our research. An example is the third row of Fig. 3 in our main paper, where our method and PerVFI predict basically the same location but the ground truth location is different. One possible explanation is that the vehicle is accelerating, but the location that our method and PerVFI predicts is based on a constant speed. As a result, at the first half of the time interval between I_0, I_1, the vehicle is slow so the ground truth is closer to the first frame, while our method and PerVFI make it approximately right in the middle.

3. Additional Results

3.1. Results in PSNR/SSIM

We include the results in PSNR/SSIM on our selected datasets in Tab. 1. We can see that PSNR/SSIM tends to be unstable and not correlated to visual qualities for methods in 2024. For example, our method underperforms Consec. BB in Xiph [9] dataset but our visual comparisons and LPIPS/FloLPIPS/FID indicate that our method is better. Similarly, PerVFI underperforms Consec. BB in Xiph-2K, but its LPIPS/FLoLPIPS/FID is much better.

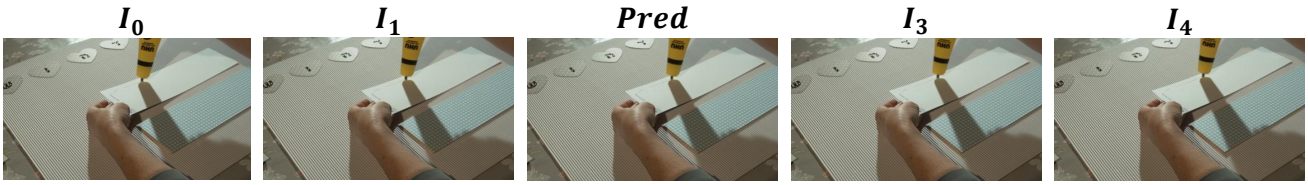
3.2. Additional Qualitative Results

More Input Frames. Our method is highly flexible and can take more than three input frames in one forward call. An example is shown in Fig. 1, where our model can receives five frames I_0, I_1, I_2, I_3, I_4 and treat either I_2 or I_1, I_3 as target. This enables us to do only one run of the sampling process for the second scenario, where LDMVFI [2] and Consec. BB [8] needs to sample twice. To achieve this, we only need to replace the second and fourth frames by zeros and send the video clip to the autoencoder, and the diffusion model only needs one sampling process to obtain latents for the decoder. However, LDMVFI and Consec. BB needs to sample frame 2 and 4 separately.

Additional Visual Comparisons. We include additional visual comparisons in Fig. 3 and Fig. 4, where examples are selected from SNU-FILM extreme [1], DAVIS [10], and Xiph-4K [9]. Our method achieves the best visual quality. Other methods exhibit distortion, blurring, or artifacts in their generation, but our method does not. Red circles and squares emphasize the area where our method achieves better quality. We encourage reviewers to do 500% zoom-in to see the results as many results contain multiple details in one frame.

8× Interpolation. $8\times$ interpolation is interpolating 7 frames between I_0, I_1, which can be done iteratively. When motion change is large, $2\times$ interpolation does not provide a good video clip and therefore we need to interpolate more frames. We include two examples of $8\times$ interpolation results in Fig. 5 just for reference. It is better to visualize $8\times$ with videos, and therefore we include more examples compared with more methods in our Project Page. $8\times$ interpolation is to interpolate 7 intermediate frames between I_0 and I_1 (i.e. only first and last frames are provided), which can be done iteratively. The upper example is taken from DAVIS [10] and the latter one from Xiph-4K [9]. The Project Page contains results from SNU-FILM extreme [1], DAVIS [10], and Xiph-4K [9]. In the upper examples of Fig. 5, bicycle tires interpolated by PerVFI [15] are missing in some frames. In the lower example, the woman's right eye becomes an artifact in PerVFI.

Flow visualization. We visualize the optical flow from interpolated results ($\hat{I}_n$) to neighboring frames I_0, I_1, shown in Fig. 2. Our primary contribution is the temporal aware latent Brownian Bridge diffusion framework instead



(a) Given two frames before and two frames next, predict the intermediate frame.

(b) Given three frames, predict frames between each adjacent frame pair.

Figure 1. (a) Given four neighboring frames I_0, I_1, I_3, I_4, we can predict the intermediate frame I_2. (b) Given a sequence of frames I_0, I_2, I_4, we can predict the intermediate frame between each adjacent pair I_1, I_3.

Table 1. Quantitative results (PSNR/SSIM) on test datasets (the higher the better).OOM indicates that the inference with one image exceeds the 24GB GPU memory of an Nvidia RTX A5000 GPU.

Methods	Xiph-4K	Xiph-2K	DAVIS	SNU-FILM			
				easy	medium	hard	extreme
	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
MCVD'22 [14]	OOM	OOM	18.946/0.705	22.201/0.828	21.488/0.812	20.314/0.766	18.464/0.694
VFIformer'22 [7]	OOM	OOM	26.241/0.850	40.130/0.991	36.090/0.980	30.670/0.938	25.430/0.864
IFRNet'22 [5]	33.970/0.943	36.570/0.966	27.313/0.877	40.100/0.991	36.120/0.980	30.630/0.937	25.270/0.861
AMT'23 [6]	34.653/0.949	36.415/0.967	27.234/0.877	39.880/0.991	36.120/0.981	30.780/0.939	25.430/0.865
UPR-Net'23 [3]	33.647/0.946	36.749/0.967	26.894/0.870	40.440/0.991	36.290/0.980	30.860/0.938	25.630/0.864
EMA-VFI'23 [16]	34.698/0.948	36.935/0.967	27.111/0.871	39.980/0.991	36.090/0.980	30.940/0.939	25.690/0.866
LDMVFI'24 [2]	OOM	OOM	25.073/0.819	38.890 0.988	33.975/0.971	29.144/0.911	23.349 0.827
PerVFI'24 [15]	32.395/0.926	34.741/0.953	26.502/0.866	38.065/0.986	34.588/0.973	29.821/0.928	25.033/0.854
Consec.BB [8]	32.153/0.927	34.964/0.956	26.391/0.858	39.637/0.990	34.886/0.974	29.615/0.929	24.376/0.848
Ours	32.441/0.928	35.748/0.959	26.272/0.860	39.460/0.990	35.308/0.977	29.529/0.929	24.513/0.847

of advancements in optical flows in other works [6, 15, 16], so we directly adopt the flow estimation architecture from Consec. BB [8]. Though the flow estimator is the same architecture as Consec. BB, our temporal design can implicitly improve it through back-propagation (see first and third row).

4. Additional Details

4.1. Implementation Details

Flow Estimator. Optical flow estimation is not our research purpose, so we use the same architecture of flow estimator in Consec. BB [8] and trained together with our autoencoder. The code for differentiable warping is available at [7, 8].

Autoencoder. The autoencoder is based on the VQ version of LDM [11]. It consists of 5 levels of image encoder and decoder, resulting in a 32× downsampling rate. Image decoders contain output channels of 64,128,128,128,256, respectively (reverse for decoder). Between the image encoder and decoder, there are four 3D convolutions with spatiotemporal attention (the last one is cross-attention) [13], where a VQ-Layer is inserted after the second 3D conv + attention. The channel dimension is 256. The VQ-Layer quantizes features into 3 channels. To predict masks M and residual Δ, we use sigmoid activation to normalize the output. The autoencoder is trained with Adam optimizer [4] and a learning rate of 10^{-5} for 35 epochs. The training loss is L1 loss and LPIPS loss, following LDM.

Brownian Bridge Diffusion. The Brownian Bridge Diffusion is implemented with 3D denoising U-Net [11] with channel dimension 32 and 3 downsample blocks as well as 3 upsample blocks, where the optimizer is Adam [4] with a learning rate of 10^{-4} and the model is trained for 50 epochs. The T for the diffusion process is set to 2. The training loss is MSE loss.

The training algorithm is shown in Algorithm 1, and the sampling algorithm is shown in Algorithm 2.

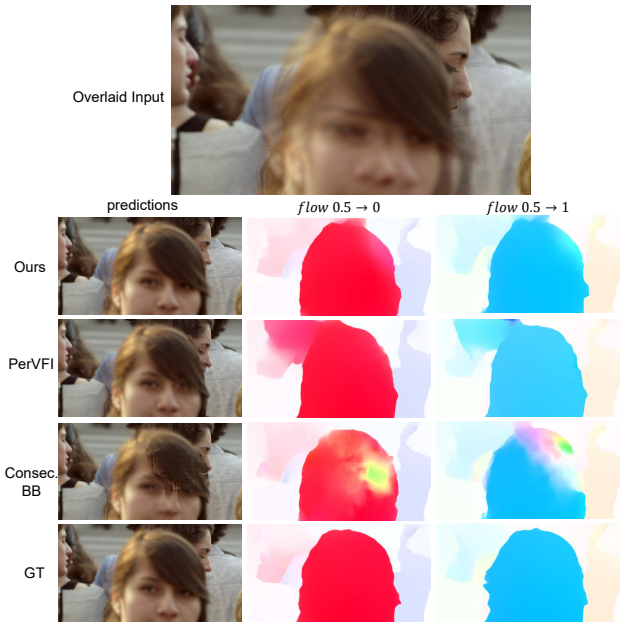


Figure 2. Visualization of optical flows.

Algorithm 1 Diffusion Training Algorithm

Let $\mathcal{E}$ be the encoder part of our autoencoder.
for $i = 1$ to $N_{training\ steps}$ **do**
  Sample $t \sim ContinuousUniform(0, T)$.
  Sample $[I_0, I_n, I_1]$ from Dataset.
  $x_0 = \mathcal{E}([I_0, I_n, I_1])$.
  Compute x_t with Eq. (5) in the main paper.
  Take gradient step on $MSE(\epsilon_\theta(x_t), x_t - x_0)$
end for

Algorithm 2 Diffusion Sampling Algorithm

Let $\mathcal{E}$ be the encoder part of our autoencoder.
Initialize $t = T, x_t = \mathcal{E}([I_0, 0, I_1])$
while $t > 0$ **do**
  Predict $x_t - x_0$ with $\epsilon_\theta(x_t)$
  Sample x_s with Eq. (6) in the main paper.
  $t \leftarrow s, x_t \leftarrow x_s$
end while

4.2. 3D Wavelet Details

The 3D wavelet transform can be considered as a convolution layer with two types of filter: high-pass filter $[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]$ and low-pass filter $[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]$.The input videos (with 3 frames) are converted to grayscale, and filters are applied in height, width, and temporal dimensions respectively. There are 8 different combinations of filters: $[HHH, HHL, HLH, HLL, LHH, LHL, LLH, LLL]$, corresponding to height, width, and temporal dimensions. We use "same padding" along height and width to keep the feature map size unchanged and use "no padding" along the temporal dimension, both with a stride of 1. Therefore, after the first extraction, it provides 8 different feature maps, where each feature map contains 2 temporal channels (the convolution reduces the temporal dimension by 1). The LLL is further extracted, resulting in 8 feature maps with only 1 temporal dimension. Feature maps other than LLL are concatenated in the temporal dimension, and we consider the temporal dimension as "channels" for model input to extract latent features. Therefore, we have a feature map with 21 channels as input.

4.3. Proof

We include the proof of our proposition in Sec. 3.3 of the main paper here:

Proof. If H_0 is rejected, then it is statistically significant to conclude that $\mathbf{x}_T - \mathbf{x}_0 \neq 0$. Therefore, we can use a simple induction to prove this. Recall that we have the diffusion and sampling processes defined as:

$$q(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_T) = \mathcal{N}\left(\frac{t}{T}\mathbf{x}_0 + (1 - \frac{t}{T})\mathbf{x}_T, \frac{t(T-t)}{T}\mathbf{I}\right). \quad (1)$$

$$p_\theta(\mathbf{x}_s|\mathbf{x}_t, \mathbf{x}_T) = \mathcal{N}\left(\mathbf{x}_t - \frac{\Delta_t}{t}(\mathbf{x}_t - \mathbf{x}_0), \frac{s\Delta_t}{t}\mathbf{I}\right). \quad (2)$$

1. Based on Eq. (2) in the main paper, suppose that at a given time step t, $\mathbf{x}_t - \mathbf{x}_0 \neq 0$, then the expectation of sampled latent at any previous step s is $\mathbb{E}(\mathbf{x}_s|\mathbf{x}_t) = \frac{s}{t}(\mathbf{x}_t - \mathbf{x}_0) + \mathbf{x}_0$.
2. By the inductive assumption, $\mathbb{E}(\mathbf{x}_s|\mathbf{x}_t) = \mathbf{x}_0 + \delta$, where $\delta \neq 0 \iff \mathbb{E}(\mathbf{x}_s|\mathbf{x}_t) \neq \mathbf{x}_0$.
3. Then, on expectation, we conclude that $\mathbf{x}_s|\mathbf{x}_t \neq \mathbf{x}_0$. Note that this is especially important in DDIM [12] sampling because the variance term is removed, in which case we can directly conclude that $\mathbf{x}_s|\mathbf{x}_t \neq \mathbf{x}_0$ without expectation.
4. Therefore, we can prove this proposition by the above induction because the sampling process is discretized into finite steps.

As a result, the sampling process is not an identity map. □

On the other hand, the sampling process is trivial because it does not change the expectation, which can be achieved with an identity map.

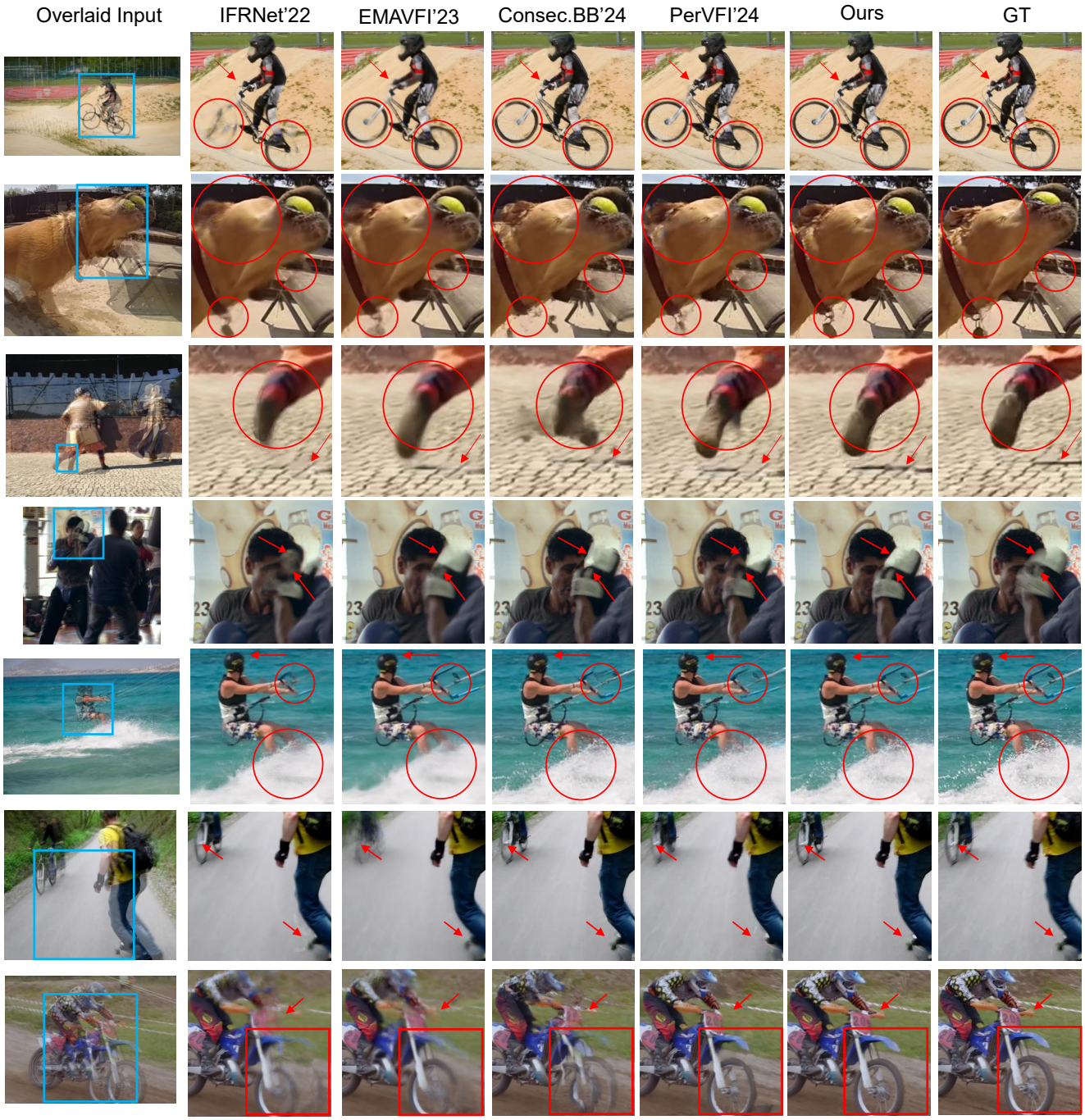


Figure 3. **Additional qualitative comparison between our method and recent SOTAs.** The leftmost image is the overlaid image of I_0 and I_1 (blended image). Images inside blue boxes contain drastic motion changes and are cropped out to show details of interpolation results. Red circles, boxes, and arrows indicate the area where we significantly perform better. Our method achieves better visual quality than recent SOTAs.

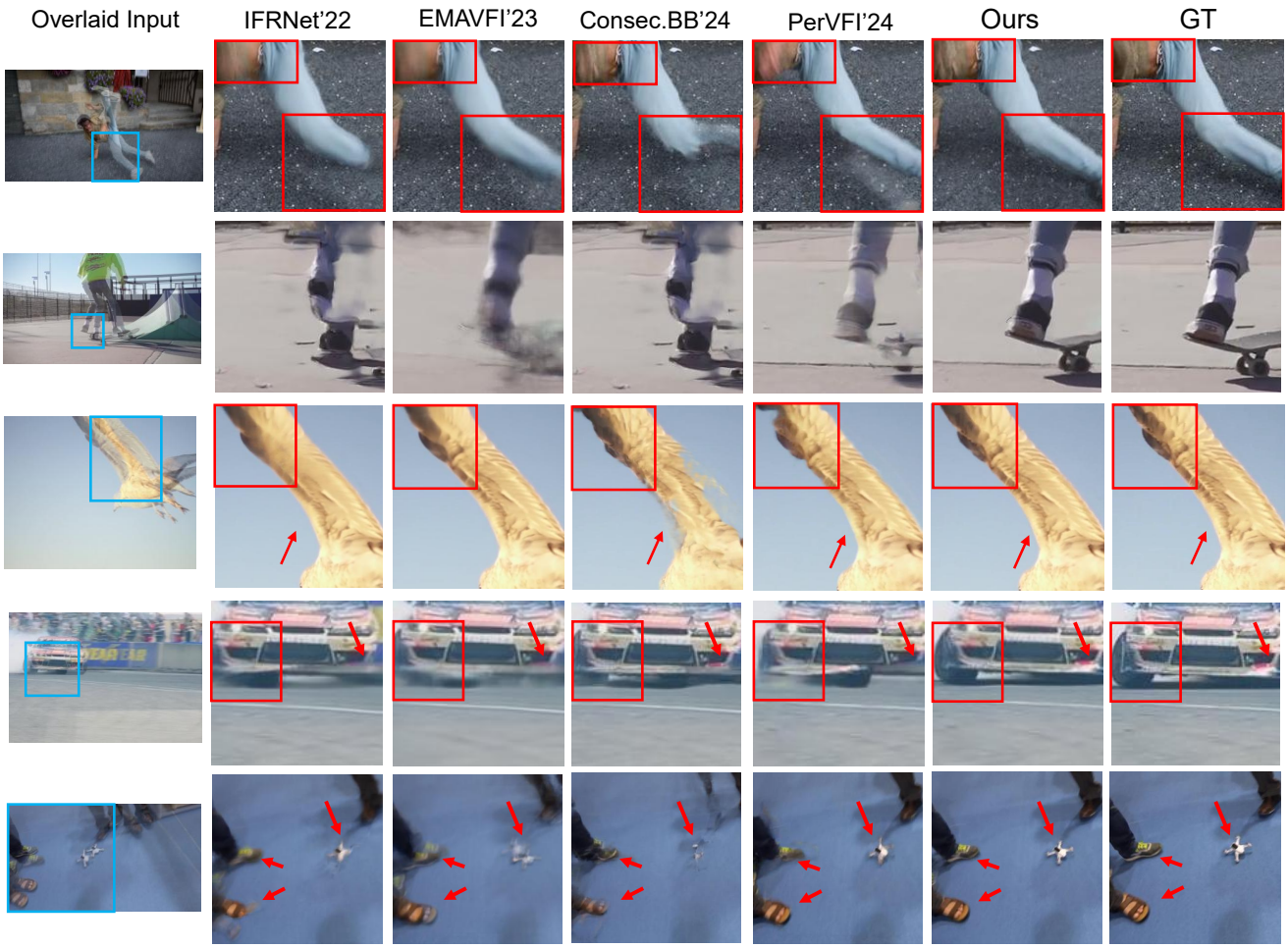


Figure 4. **Additional qualitative comparison between our method and recent SOTAs.** The leftmost image is the overlaid image of I_0 and I_1 (blended image). Images inside blue boxes contain drastic motion changes and are cropped out to show details of interpolation results. Red circles, boxes, and arrows indicate the area where we significantly perform better. Our method achieves better visual quality than recent SOTAs.

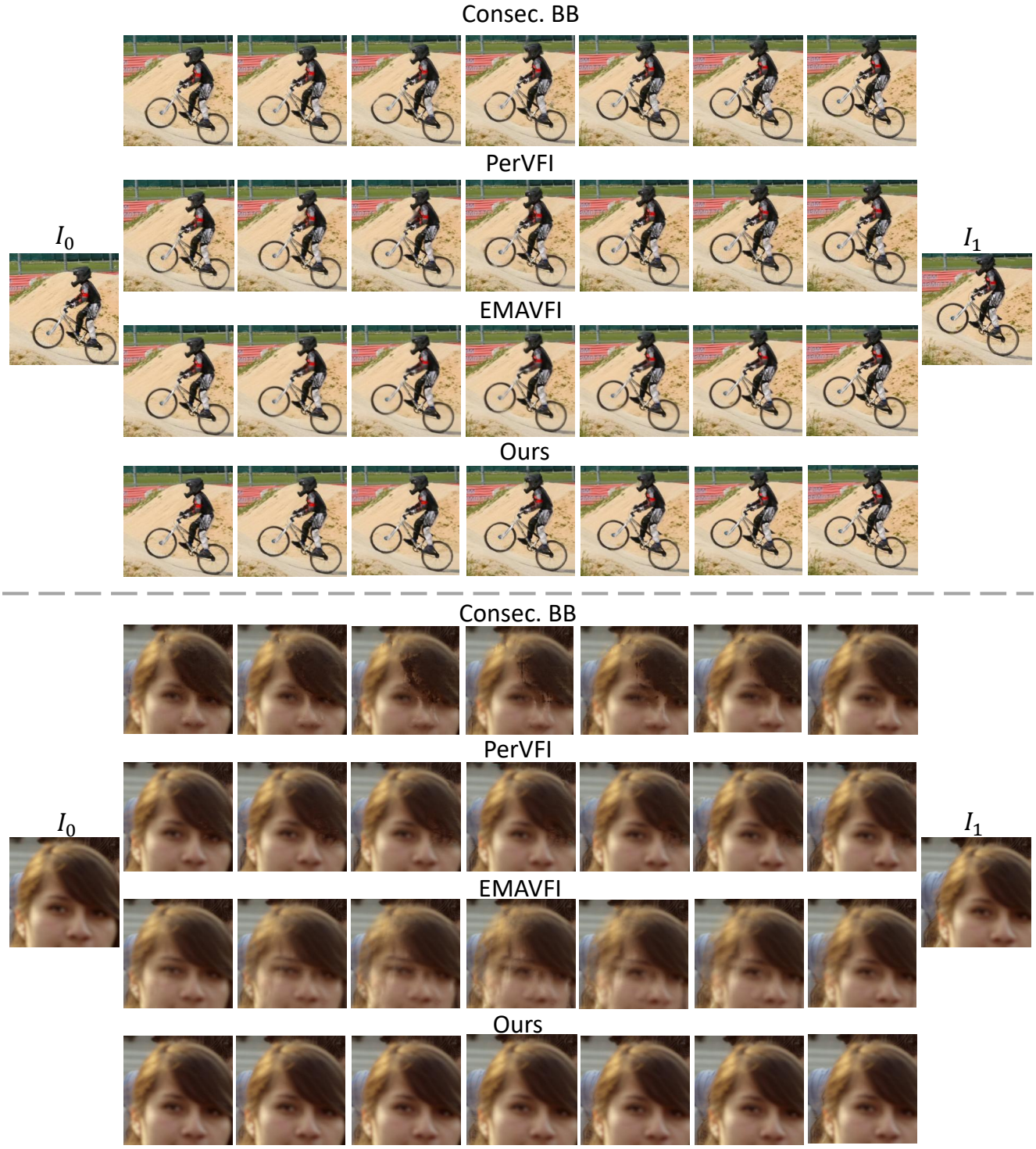


Figure 5. **Visual comparison of 8x× interpolation results.** We include a visual comparison of 8× interpolation between our method and PerVFI. Red arrows indicate where our method is visually better. Additional comparisons (in video form) are provided in our Project Page.

References

[1] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 1

[2] Duolikun Danier, Fan Zhang, and David Bull. Ldmvfi: Video frame interpolation with latent diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1472–1480, 2024. 1, 2

[3] Xin Jin, Longhai Wu, Jie Chen, Youxin Chen, Jayoon Koo, and Cheul-hee Hahm. A unified pyramid recurrent network for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 2

[5] Lingtong Kong, Boyuan Jiang, Donghao Luo, Wenqing Chu, Xiaoming Huang, Ying Tai, Chengjie Wang, and Jie Yang. Ifrnet: Intermediate feature refine network for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[6] Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. Amt: All-pairs multi-field transforms for efficient frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[7] Liying Lu, Ruizheng Wu, Huaijia Lin, Jiangbo Lu, and Jiaya Jia. Video frame interpolation with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[8] Zonglin Lyu, Ming Li, Jianbo Jiao, and Chen Chen. Frame interpolation with consecutive brownian bridge diffusion. *arXiv preprint arXiv:2405.05953*, 2024. 1, 2

[9] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1

[10] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[11] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022. 2

[12] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 3

[13] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 2

[14] Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in neural information processing systems*, 2022. 2

[15] Guangyang Wu, Xin Tao, Changlin Li, Wenyi Wang, Xiaohong Liu, and Qingqing Zheng. Perception-oriented video frame interpolation via asymmetric blending. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2753–2762, 2024. 1, 2

[16] Guozhen Zhang, Yuhan Zhu, Haonan Wang, Youxin Chen, Gangshan Wu, and Limin Wang. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2