

Frame Interpolation with Consecutive Brownian Bridge Diffusion (Supplementary Material)

1 OVERVIEW OF SUPPLEMENTARY MATERIAL

The supplementary material is organized into the following sections:

- Section 2: Detailed formula derivation.
- Section 3: Connection between our method and diffusion SDEs [13].
- Section 4: Additional experimental results, including evaluation with a better-converged autoencoder, results in PSNR/SSIM, additional visualizations, and additional ablation studies.

2 FORMULA DERIVATION

2.1 Consecutive Brownian Bridge

For $0 < t < h$, if we have $s > t$, then the Markov property of the Wiener process produces:

$$W_s|(W_0, W_t, W_h) = W_s|(W_t, W_h)$$

Applying in our setting, this becomes: $W_t|W_T = \mathbf{x}$, $W_{2T} = \mathbf{z}$ for $t > T$. Note that only the variance of the Wiener process is related to time, and the variance of general Brownian Bridge $W_t|(W_{t_1}, W_{t_2})$ is $\frac{(t_2-t)(t-t_1)}{t_2-t_1}$. If we add any value simultaneously to t_1, t_2, t , the variance is unchanged. Therefore, we can subtract T in time to get $W_s|W_0 = \mathbf{x}$, $W_T = \mathbf{z}$, where $s = t - T$.

If we have $s < t$, then it is important to know that tW_{t-1} is a Wiener process with the same distribution with W_t [9]. We can simply add a small ϵ to time and use such transformation to obtain:

$$\begin{aligned} W_s|(W_0, W_t, W_h) &= W_{s+\epsilon}|(W_\epsilon, W_{t+\epsilon}, W_{h+\epsilon}) \\ &= (s+\epsilon)W_{(s+\epsilon)^{-1}}|\epsilon W_{\epsilon^{-1}}, (t+\epsilon)W_{(t+\epsilon)^{-1}}, (h+\epsilon)W_{(h+\epsilon)^{-1}} \\ &= (s+\epsilon)W_{(s+\epsilon)^{-1}}|\epsilon W_{\epsilon^{-1}}, (t+\epsilon)W_{(t+\epsilon)^{-1}} \\ &= W_s|(W_0, W_t) \end{aligned}$$

In our method, this becomes $W_t|W_0 = \mathbf{y}$, $W_T = \mathbf{x}$. The distribution is $\mathcal{N}(\frac{t}{T}\mathbf{y} + (1 - \frac{t}{T})\mathbf{x}, \frac{t(T-t)}{T}\mathbf{I})$. Now, let's consider another process defined as $W_s|W_0 = \mathbf{x}$, $W_T = \mathbf{y}$. The distribution is easy to derive: $\mathcal{N}(\frac{s}{T}\mathbf{x} + (1 - \frac{s}{T})\mathbf{y}, \frac{s(T-s)}{T}\mathbf{I})$. With simple algebra, we can find that when $s = T - t$, the two distributions are equal. Thus, we finish the derivation of the distribution of consecutive Brownian Bridge.

Unpublished working draft. Not for distribution.

Permission to make digital or hard copies of all or part of this work for personal or classroom use, or to copy and redistribute to peers, is granted by ACM, provided that the copies are made without charge and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or to publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ACM Multimedia '24, October 28– November 01, 2024, Melbourne, Australia
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/18/06
<https://doi.org/XXXXXXX.XXXXXXX>

2.2 Cumulative Variance

We denote \mathbf{z} as standard Gaussian distribution. In DDPM [4], $\mathbf{x}_{t-1} = \frac{1}{\sqrt{1-\beta_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}} \epsilon_\theta \right) + \sqrt{\beta_t} \mathbf{z}$. At the first step of generation, since $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $0 < \beta_t < 1$, we have:

$$\begin{aligned} Var(\mathbf{x}_{T-1}) &= Var \left(\frac{1}{\sqrt{1-\beta_t}} \left(\mathbf{x}_T - \frac{\beta_t}{\sqrt{1-\alpha_t}} \epsilon_\theta \right) + \sqrt{\beta_t} \mathbf{z} \right) \\ &> Var \left(\frac{1}{\sqrt{1-\beta_t}} \mathbf{x}_T + \sqrt{\beta_t} \mathbf{z} \right) \\ &> 1 + \beta_t \end{aligned}$$

Since ϵ_θ takes random input, it has a positive variance. The following sampling steps have fixed inputs \mathbf{x}_t , so the variance only contains β_t . Therefore, the cumulative variance is larger than $1 + \sum_t \beta_t$, corresponding to **11.036** in real experiments. However, in our method, we have $\mathbf{x}_{t-\Delta_t} = \mathbf{x}_t - \frac{\Delta_t}{t} \epsilon_\theta + \sqrt{\frac{(t-\Delta_t)\Delta_t}{t}} \mathbf{z}$, and \mathbf{x}_T is deterministic, we have:

$$\begin{aligned} Var(\mathbf{x}_{t-\Delta_t}) &= Var \left(\mathbf{x}_t - \frac{\Delta_t}{t} \epsilon_\theta + \sqrt{\frac{(t-\Delta_t)\Delta_t}{t}} \mathbf{z} \right) \\ &= Var \left(\sqrt{\frac{(t-\Delta_t)\Delta_t}{t}} \mathbf{z} \right) \\ &< \Delta_t \end{aligned}$$

Since ϵ_θ takes fixed inputs, it has no variance. The cumulative variance is smaller than $\sum_t \Delta_t = T$, corresponding to **2** in our experiments. We mentioned this result in Section 3.4 in our main paper.

3 CONNECTION WITH DIFFUSION SDES

Our method can be easily written in score-based SDE [1, 13, 18]. The forward process of score-based SDEs is defined as:

$$d\mathbf{x} = f(\mathbf{x}, t)dt + g(t)d\mathbf{w}. \quad (1)$$

$f(\mathbf{x}, t)$ is the drift term, and $g(t)$ is the dispersion term. \mathbf{w} denotes the standard Wiener process. The corresponding reversed SDE is defined as:

$$d\mathbf{x} = [f(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})] dt + g(t)d\tilde{\mathbf{w}}. \quad (2)$$

The conditional generation counterpart is defined as:

$$d\mathbf{x} = \{f(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} [\log p_t(\mathbf{x}) + \log p_t(\mathbf{y}|\mathbf{x})]\} dt + g(t)d\tilde{\mathbf{w}}. \quad (3)$$

The term \mathbf{y} is the conditional control for generation. Moreover, there exists a deterministic ODE trajectory (probability flow ODE) with the same marginal distribution $p_t(\mathbf{x})$ with Eq. (2) [13]:

$$d\mathbf{x} = \left[f(\mathbf{x}, t) - \frac{1}{2} g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right] dt. \quad (4)$$

Therefore, it suffices to train a neural network s_θ estimating $\nabla_x \log p_t(\mathbf{x})$ [13]. Indeed, Brownian Bridge can be written in SDE form by [9]:

$$d\mathbf{x} = \frac{\mathbf{y} - \mathbf{x}_t}{T - t} dt + d\mathbf{w}. \quad (5)$$

\mathbf{y} is another endpoint of the Brownian Bridge. The reversed SDE is defined as:

$$d\mathbf{x} = \left[\frac{\mathbf{y} - \mathbf{x}_t}{T - t} - \nabla_x \log p_t(\mathbf{x}) \right] dt + d\bar{\mathbf{w}}. \quad (6)$$

By our formulation, our proposed method is compatible with score-based SDEs. Moreover, compared with conditional SDEs in Eq. (3), this formulation does not include $\log p_t(\mathbf{y}|\mathbf{x})$ which needs estimation.

4 ADDITIONAL RESULTS

4.1 Better-Converged Autoencoder

As we claimed in our main paper, the autoencoder is still converging, so we train the autoencoder with an additional 80 epochs to evaluate the performance. The updated results are shown in Table 2 and Table 3, corresponding to Table 1 and Table 2 in our main paper. After additional training, our method achieves the best performance except for FID in the SNU-FILM extreme subset [2] and LPIPS in UCF-101 [14]. It is important to note that our autoencoder can still further converge, and the architecture of the autoencoder is not optimized (our main focus is the diffusion model rather than the autoencoder). Our performance on the DAVIS dataset [11] gets slightly degraded in FloLPIPS and FID while improved in LPIPS. It might be because our method did not improve on the DAVIS dataset, but the weight-changing of our model makes the performance slightly vary.

4.2 Quantitative Results

We provide the evaluation results (with the latest weights in Section 4.1) in PSNR/SSIM in Table 4. Though our method does not have a good performance in PSNR/SSIM, it is due to the **inconsistency** between PSNR/SSIM and visual quality (see Section 4.3 and Figure 1). Therefore, we choose LPIPS/FloLPIPS/FID as our main evaluation metrics.

4.3 Qualitative Results

Inconsistency Between PSNR/SSIM and Visual Quality. We provide some examples to demonstrate the inconsistency between PSNR/SSIM and visual quality, as shown in Figure 1. Our method achieves better visual quality than UPR-Net [5] such as clearer dog skins, clearer cloth with folds, and clearer shoes and fences with nets. However, we did not achieve a satisfactory PSNR/SSIM, which is 5-10% lower than that of UPR-Net.

Additional Qualitative Comparisons. In addition, we provide more qualitative comparisons between our method and LDMVFI [3] in Figure 3 and qualitative comparisons between our method and recent SOTAs in Figure 4. All examples are selected from SNU-FILM extreme [2].

Multi-frame Interpolation. We provide qualitative results of multi-frame interpolation of our methods and LDMVFI [3]. Multi-frame interpolation is achieved in a bisection manner. We first interpolate $I_{0.5}$ with I_0, I_1 , and then we interpolate $I_{0.25}$ with $I_0, I_{0.5}$ and

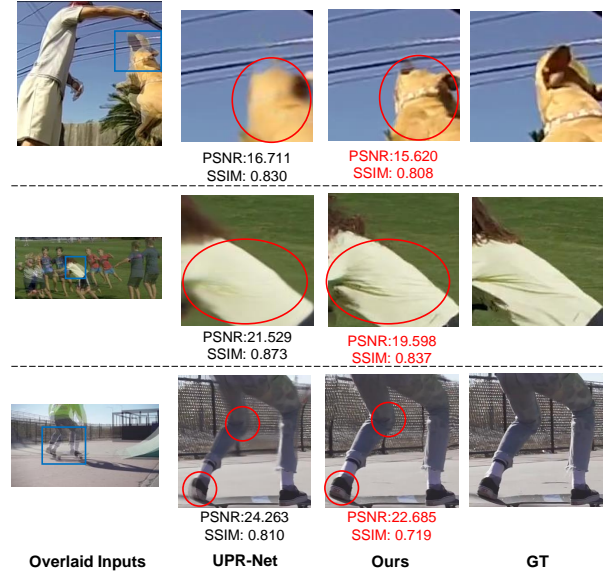


Figure 1: Visual illustration of the inconsistency between PSNR/SSIM and visual quality. Only images cropped within blue boxes are evaluated with PSNR/SSIM. The red circles highlight our visual quality. Our method generates images with better visual quality, but the PSNR/SSIM is much lower.

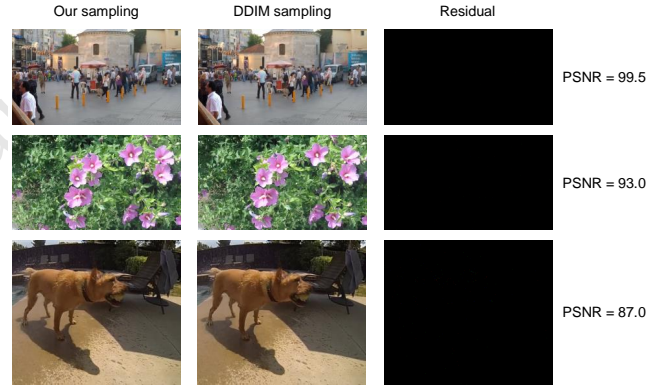


Figure 2: Visual comparison between our sampling and DDIM sampling with 5 steps generation. They achieve almost identical results (with very large PSNR). The residual is the absolute difference between the two images. Black means 0 difference, and almost everywhere is black.

$I_{0.75}$ with $I_{0.5}, I_1$. More frames can be interpolated in this manner. We interpolate 7 frames between two I_0, I_1 , and the visual comparisons are presented in Figure 5. All examples are selected from SNU-FILM hard [2]. Additional video demos are shown on an anonymous GitHub page: <https://anonymous.4open.science/w/interpolation/>. Due to the bisection-like multi-frame interpolation method, the multi-frame interpolation results largely depends on the first step of interpolation ($I_{0.5}$). If $I_{0.5}$ achieves good quality, then the relative motion in the second step (interpolating $I_{0.25}, I_{0.75}$) is easy to achieve high quality because the motion changes become smaller.

Table 1: Ablation study on the number of sampling steps. This experiment is conducted on SNU-FILM extreme subset [2].

Number of steps	LPIPS	FloLPIPS	FID
200	0.074	0.075	41.264
100	0.074	0.075	41.265
50	0.074	0.075	41.264
20	0.074	0.075	41.266
5	0.074	0.075	41.264

However, if the interpolation quality is not good at the first step, then later steps will not achieve good quality because such an unsatisfactory quality will be transmitted. LDMVFI 3 tends to generate overlaid or distorted $I_{0.5}$, resulting in unsatisfactory multi-frame interpolation results. [We largely alleviate this problem, resulting in much better and more realistic interpolated videos.](#)

4.4 Ablation Studies

Number of Sampling Steps. We investigate how the number of sampling steps will impact the performance. This ablation study is conducted on SNU-FILM extreme subset [2], shown in Table 1. We observe that the performance remains almost identical. The reason could be the relatively small differences between neighboring frames. Our method does not convert random noise to images like DDPM [4]. Instead, we convert one image to its neighboring frames, so we do not need to generate details from random noises. Instead, we change details from existing details, and therefore it may not need many steps to generate.

DDIM Sampling. As we claimed, our formulation does not need DDIM [12] sampling to accelerate. We compare our sampling with DDIM sampling with $\eta = 0$ in 5 sampling steps for comparison (evaluated with the latest weights). The visual result is shown in Figure 2. There is almost no difference between the output of our sampling method and DDIM sampling, indicating that we do not require such a method to accelerate sampling.

Table 2: Quantitative results (LPIPS/FloLPIPS/FID, the lower the better) on test datasets. † means we evaluate our consecutive Brownian Bridge diffusion (trained on Vimeo 90K triplets [16]) with autoencoder provided by LDMVFI [3]. The best performances are boldfaced, and the second best performances are underlined.

Methods	Middlebury	UCF-101	DAVIS	SNU-FILM			
				easy	medium	hard	extreme
	LPIPS/FloLPIPS/FID	LPIPS/FloLPIPS/FID	LPIPS/FloLPIPS/FID	LPIPS/FloLPIPS/FID	LPIPS/FloLPIPS/FID	LPIPS/FloLPIPS/FID	LPIPS/FloLPIPS/FID
ABME'21 [10]	0.027/0.040/11.393	0.058/0.069/37.066	0.151/0.209/16.931	0.022/0.034/6.363	0.042/0.076/15.159	0.092/0.168/34.236	0.182/0.300/63.561
MCVD'22 [15]	0.123/0.138/41.053	0.155/0.169/102.054	0.247/0.293/28.002	0.199/0.230/32.246	0.213/0.243/37.474	0.250/0.292/51.529	0.320/0.385/83.156
VFIformer'22 [8]	0.015/0.024/9.439	0.033/0.040/22.513	0.127/0.184/14.407	0.018/0.029/5.918	0.033/0.053/11.271	0.061/0.100/22.775	0.119/0.185/40.586
IFRNet'22 [6]	0.015/0.030/10.029	0.029/0.034/20.589	0.106/0.156/12.422	0.021/0.031/6.863	0.034/0.050/12.197	0.059/0.093/23.254	0.116/0.182/42.824
AMT'23 [7]	0.015/0.023/7.895	0.032/0.039/21.915	0.109/0.145/13.018	0.022/0.034/6.139	0.035/0.055/11.039	0.060/0.092/20.810	0.112/0.177/40.075
UPR-Net'23 [5]	0.015/0.024/7.935	0.032/0.039/21.970	0.134/0.172/15.002	0.018/0.029/5.669	0.034/0.052/10.983	0.062/0.097/22.127	0.112/0.176/40.098
EMA-VFI'23 [17]	0.015/0.025/8.358	0.032/0.038/21.395	0.132/0.166/15.186	0.019/0.038/5.882	0.033/0.053/11.051	0.060/0.091/20.679	0.114/0.170/39.051
LDMVFI'24 [3]	0.019/0.044/16.167	0.026/0.035/26.301	0.107 0.153/12.554	0.014/0.024/5.752	<u>0.028/0.053/12.485</u>	0.060/0.114/26.520	0.123/0.204/47.042
Ours†	<u>0.012/0.011/14.447</u>	0.030/0.029/15.335	0.097/0.145/12.623	0.011/0.011/5.737	0.028/0.028/12.569	0.051/0.053/25.567	0.099/0.103/46.088
Ours	0.007/0.008/7.493	<u>0.029/0.028/13.898</u>	0.051/0.090/10.190	0.009/0.009/4.992	0.021/0.022/9.301	0.034/0.035/19.852	0.074/0.074/41.264

Table 3: Ablation studies of autoencoder and ground truth estimation. + GT means we input ground truth x to the decoder part of autoencoder. + BB indicates our consecutive Brownian Bridge diffusion trained with autoencoder of LDMVFI. With our consecutive Brownian Bridge diffusion, the interpolated frame has almost the same performance as the interpolated frame with ground truth latent representation, indicating the strong ground truth estimation capability. Our autoencoder also has better performance than LDMVFI [3].

Methods	Middlebury	UCF-101	DAVIS	SNU-FILM			
				easy	medium	hard	extreme
	LPIPS/FloLPIPS/FID	LPIPS/FloLPIPS/FID	LPIPS/FloLPIPS/FID	LPIPS/FloLPIPS/FID	LPIPS/FloLPIPS/FID	LPIPS/FloLPIPS/FID	LPIPS/FloLPIPS/FID
LDMVFI'24 [3]	0.019/0.044/16.167	0.026/0.035/26.301	0.107 0.153/12.554	0.014/0.024/5.752	0.028/0.053/12.485	0.060/0.114/26.520	0.123 0.204/47.042
LDMVFI'24 [3] + BB	0.012/0.011/14.447	0.030/0.029/15.335	0.097/0.145/12.623	0.011/0.011/5.737	0.028/0.028/12.569	0.051/0.053/25.567	0.099/0.103/46.088
LDMVFI'24 [3] + GT	0.012/0.011/14.492	0.030/0.029/15.338	0.097/0.145/12.670	0.011/0.011/5.738	0.028/0.028/12.574	0.051/0.053/25.655	0.099/0.103/46.080
Ours	0.007/0.008/7.493	0.029/0.028/13.898	0.051/0.090/10.190	0.009/0.009/4.992	0.021/0.022/9.301	0.034/0.035/19.852	0.074/0.074/41.264
Ours + GT	0.007/0.008/7.486	0.029/0.028/13.898	0.051/0.090/10.189	0.009/0.009/4.994	0.021/0.022/9.230	0.034/0.035/19.850	0.074/0.074/41.265

Table 4: Quantitative results (PSNR/SSIM) on test datasets (the higher the better). † means we evaluate our consecutive Brownian Bridge diffusion (trained on Vimeo 90K [16]) with autoencoder provided by LDMVFI [3].

Methods	Middlebury	UCF-101	DAVIS	SNU-FILM			
				easy	medium	hard	extreme
	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
ABME'21 [10]	37.639/0.986	35.380/0.970	26.861/0.865	39.590/0.990	35.770/0.979	30.580/0.936	25.430/0.864
MCVD'22 [15]	20.539/0.820	18.775/0.710	18.946/0.705	22.201/0.828	21.488/0.812	20.314/0.766	18.464/0.694
VFIformer'22 [8]	38.438/0.987	35.430/0.970	26.241/0.850	40.130/0.991	36.090/0.980	30.670/0.938	25.430/0.864
IFRNet'22 [6]	36.368/0.983	35.420/0.967	27.313/0.877	40.100/0.991	36.120/0.980	30.630/0.937	25.270/0.861
AMT'23 [7]	38.395/0.988	35.450/0.970	27.234/0.877	39.880/0.991	36.120/0.981	30.780/0.939	25.430/0.865
UPR-Net'23 [5]	38.065/0.986	35.470/0.970	26.894/0.870	40.440/0.991	36.290/0.980	30.860/0.938	25.630/0.864
EMA-VFI'23 [17]	38.526/0.988	35.480/0.970	27.111/0.871	39.980/0.991	36.090/0.980	30.940/0.939	25.690/0.866
LDMVFI'24 [3]	34.230/0.974	32.160/0.964	25.073/0.819	38.890 0.988	33.975/0.971	28.144/0.911	23.349 0.827
Ours†	34.057/0.970	34.730/0.965	25.446/0.837	38.720/0.988	34.016/0.971	28.556/0.918	23.931/0.837
Ours	35.709/0.971	34.941/0.968	25.994/0.849	39.162/0.988	34.886/0.974	29.158/0.921	24.084/0.838



Figure 3: Additional Qualitative Comparison of our methods and LDMVFI. Images cropped with blue boxes are shown for better-detailed comparison. Our method steadily achieves better visual quality.

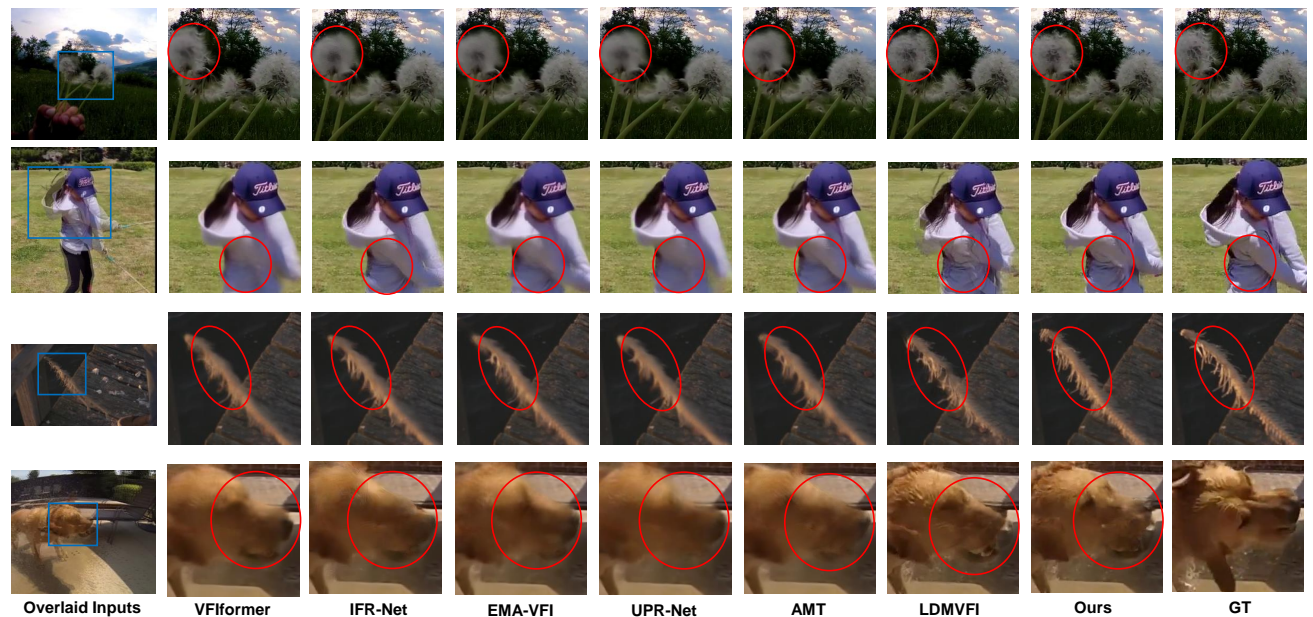


Figure 4: Additional Qualitative Comparison of our methods and recent SOTAs. Only images within the blue box are displayed for better-detailed comparison.

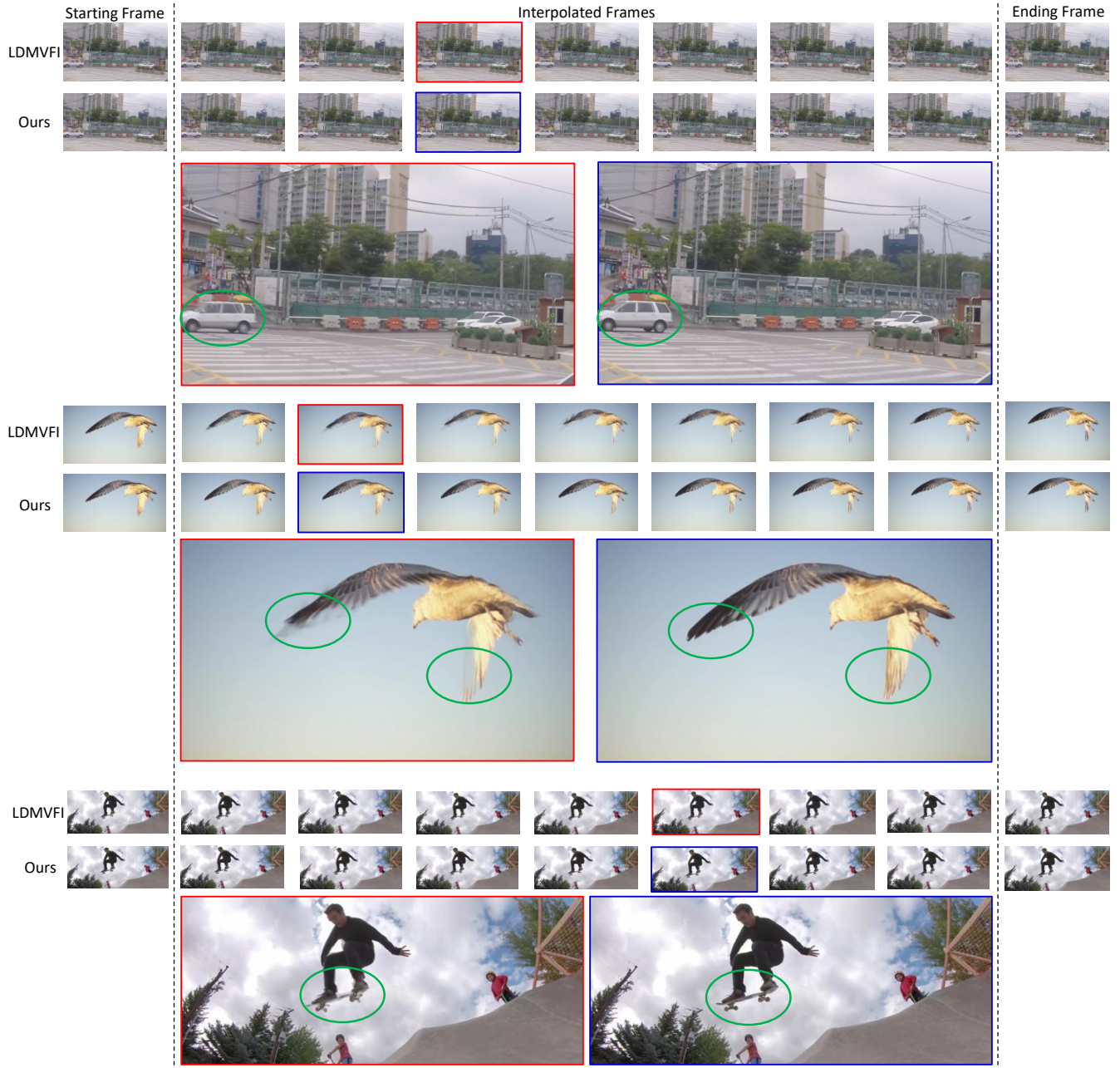


Figure 5: Multi-frame interpolation results. LDMVFI usually interpolates distorted or overlaid images while ours does not. Images with red and blue borders are displayed to show details. Our method corresponds to the blue border while LDMVFI corresponds to the red. Green circles highlight the detail where our performance is better.

REFERENCES

- [1] Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. 2021. Conditional image generation with score-based diffusion models. *arXiv preprint arXiv:2111.13606* (2021).
- [2] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. 2020. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- [3] Duolikun Danier, Fan Zhang, and David R. Bull. 2024. LDMVFI: Video Frame Interpolation with Latent Diffusion Models. In *AAAI Conference on Artificial Intelligence*.
- [4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* (2020).
- [5] Xin Jin, Longhai Wu, Jie Chen, Youxin Chen, Jayoon Koo, and Cheul-hee Hahm. 2023. A unified pyramid recurrent network for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- [6] Lingtong Kong, Boyuan Jiang, Donghao Luo, Wenqing Chu, Xiaoming Huang, Ying Tai, Chengjie Wang, and Jie Yang. 2022. IFRNet: Intermediate Feature Refine Network for Efficient Frame Interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [7] Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. 2023. AMT: All-Pairs Multi-Field Transforms for Efficient Frame Interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [8] Liying Lu, Ruizheng Wu, Huaijia Lin, Jiangbo Lu, and Jiaya Jia. 2022. Video frame interpolation with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- [9] Bernt Oksendal. 2013. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media.
- [10] Junheum Park, Chul Lee, and Chang-Su Kim. 2021. Asymmetric Bilateral Motion Estimation for Video Frame Interpolation. In *International Conference on Computer Vision*.
- [11] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. 2016. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [12] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.
- [13] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.
- [14] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012).
- [15] Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. 2022. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in neural information processing systems* (2022).
- [16] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. 2019. Video Enhancement with Task-Oriented Flow. *International Journal of Computer Vision (IJCV)* (2019).
- [17] Guozhen Zhang, Yuhao Zhu, Haonan Wang, Youxin Chen, Gangshan Wu, and Limin Wang. 2023. 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*.
- [18] Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. 2024. Denoising Diffusion Bridge Models. In *The Twelfth International Conference on Learning Representations*.