

Generative Video Bi-flow: Supplemental Materials

Chen Liu
University College London
chen.liu.21@ucl.ac.uk

Tobias Ritschel
University College London
t.ritschel@ucl.ac.uk

We provide more details about our implementation in Sec. A and additional quantitative results in Sec. B. We recommend visiting our supplemental HTML material ([index.html](#)) for more generated videos and qualitative evaluations.

A. Implementation Details

Our implementation uses PyTorch [1]. We model the ODE flow field using the UNet implementation from Huggingface Diffusers [8]. For FLOW and CONDIFF, the UNet network has seven levels of convolutional blocks, with channel sizes of 192, 192, 384, 384, 384, 768, and 768, respectively, and spatial attention layers involved in the penultimate level. For BI-FLOW, the joint field consists of two separate smaller UNet networks of the same design, except channel sizes being 128, 128, 256, 256, 256, 512, and 512 now. While we opt not to use a unified model or share parameters among the sub-networks—thereby avoiding the tedious process of tuning loss weights for the two joint losses—we still consider them as a single model to use, rather than as disjoint components. To ensure fair comparisons, we guarantee that all models have nearly equivalent efficiency and memory footprints, as demonstrated in Tab. 1.

Table 1. We benchmark the models for different methods in terms of the number of parameters (memory) and GFlops (efficiency). GFlops is measured in a 128^2 input.

Models	#Params (M)	GFlops
FLOW	277.867	134.48
BI-FLOW	247.693	119.61
CONDIFF	277.872	134.56

The training details are the same across all datasets. We train all methods using the default AdamW optimizer with a learning rate of 1×10^{-4} . The batch size is 128, and the number of training iterations is 200K. We perform gradient accumulation and low-precision training of “BF16” using Huggingface Accelerate [3] to fit the large batch size into the limited GPU memory. Our training operates directly in

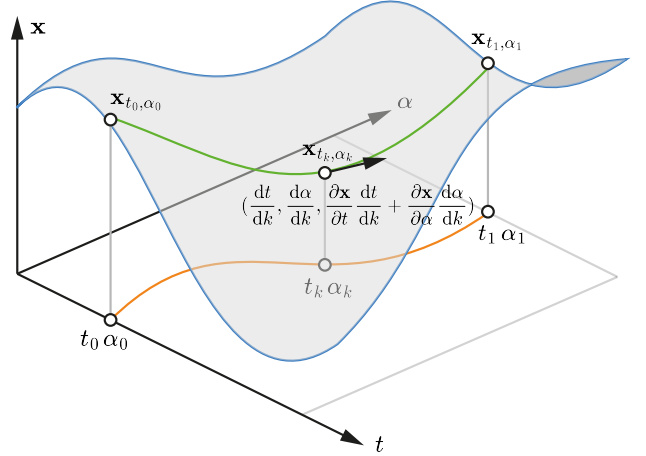


Figure 1. Characteristic ODE. The orange curve is the characteristic curve parameterized by k in the $t\alpha$ -plane. The solutions of the PDE form a manifold, over which the green curve is the corresponding ODE solving trajectory of this characteristic curve. The tangent of the green curve is the black arrow in the center, with the tangent of the orange curve as the first two dimensions and the directional derivative of \mathbf{x} as the third dimension.

RGB space. With the above setting, the average training time is approximately 80 hours in one Nvidia RTX 4090 GPU with PyTorch compilation. We use Torchode [4] to solve the learned flow field.

To further explain our joint sampling, in Fig. 1 we provide an illustration of the characteristic ODE curve (Eq. 8 in the main paper).

B. More Quantitative Results

Besides the main discussion of video quality versus speed, we report FID for realism and I-SIM for consistency to further investigate frame quality. FID is computed between all 65,536 generated frames (128 videos of 512 frames) and all frames in the test set. I-SIM is the average cosine similarity of Inception features between consecutive frames of generated videos. The closer the I-SIM is to one, the smaller the changes between frames. The numbers are shown in Tab. 2.

As expected, FLOW wins in terms of consistency, closely followed by BI-FLOW without noise added ($\epsilon = 0.0$). This

Table 2. The Fréchet inception distance (FID) and I-SIM evaluations for all methods and datasets. The methods and variants are solved using the same configuration of the adaptive solver in the main paper.

	SKY		BIKING		RIDING		CARLA		MAZES		MINERL	
	FID ↓	I-SIM ↑	FID ↓	I-SIM ↑	FID ↓	I-SIM ↑	FID ↓	I-SIM ↑	FID ↓	I-SIM ↑	FID ↓	I-SIM ↑
FLOW	246.3	0.999	337.6	0.997	282.7	0.997	287.3	0.997	283.9	0.996	228.0	0.998
BI-FLOW ($\epsilon = 0.0$)	252.9	0.998	307.9	0.993	322.8	0.993	285.7	0.993	214.0	0.995	214.8	0.998
BI-FLOW ($\epsilon = 0.1$)	70.5	0.961	211.1	0.956	115.5	0.939	46.1	0.943	97.7	0.928	89.3	0.941
BI-FLOW ($\epsilon = 0.2$)	76.7	0.951	141.7	0.918	83.6	0.916	29.1	0.931	32.6	0.883	53.9	0.900
BI-FLOW ($\epsilon = 0.3$)	80.6	0.948	121.0	0.897	70.3	0.903	26.4	0.928	29.8	0.879	44.4	0.882
CONDIFF	191.7	0.981	249.5	0.888	258.7	0.929	124.9	0.929	30.2	0.881	140.5	0.859

is because the video trajectory modeled by the ODE flow is inherently continuous and coherent, as also evidenced in recent work [5]. For BI-FLOW there is an obvious trend that FID and I-SIM metrics both decrease with increasing noise levels. The decline is particularly pronounced from a noiseless condition ($\epsilon = 0.0$) to one with noise ($\epsilon = 0.1$). This confirms our observation in the main paper that this is a trade-off spectrum where we trade frame quality with frame consistency. The lowest FID is always achieved by BI-FLOW.

Table 3. More baseline comparisons in SKY and CARLA datasets.

BI-FLOW	RIVER	SVP
436	794	909

In Tab. 3, we compare to RIVER [2] and SVP [6] in FVD. We train them in pixel space for comparison. RIVER is implemented with the official codebase and the finetuned warm-up hyperparameter. For SVP, we only adopt the modified inference as their other contributions focus on portrait video generation. It shows that our bi-flow outperforms them. With as few solving steps as bi-flow, both RIVER and SVP accumulate errors rapidly, especially when generating frames beyond the training horizon, while RIVER performs better than SVP due to the sparse conditioning. It requires approximately $2.53\times$ and $2.69\times$ steps for RIVER and SVP to achieve similar FVD as bi-flow.

We further verify that the sparse conditioning of past frames, one of RIVER’s contributions, can also be applied to our bi-flow and improve FVD from 436 to 387 (11%) by conditioning on one more random previous frame. We believe that it is orthogonal to our method and shows a promising future direction.

We show the scalability of our bi-flow in model size and training data in Fig. 2. For model size, we report the FVD achieved by four variants of the same backbone model, with relative sizes approximately $0.5\times$, $1.0\times$ (original), $2.0\times$, and $4.0\times$. To demonstrate data scaling, we report FVD on the same test set using random subsets of the training data at

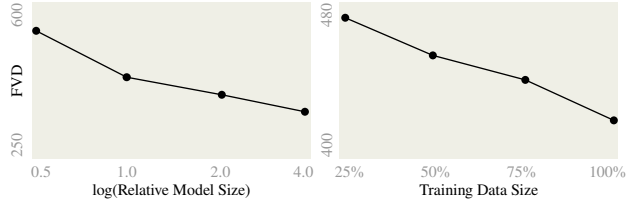


Figure 2. Scalability test.

25%, 50%, 75%, and 100% (original). The results confirm that our bi-flow can benefit from increasing model capability and data volume. We have not yet observed saturation within the limit of our available compute resources.

Table 4. UCF101 FVD results.

BI-FLOW	FLOW	CONDIFF	RIVER	SVP
4512	5666	5370	4794	5218

To make our evaluation more systematic, we add FVD results of UCF101 [7], a multi-class video datasets, in Tab. 4. The FVD results reflect the difficulty that UCF101 is significantly more challenging, particularly for unconditional generation. Video bi-flow continues to outperform the baselines in this situation.

References

- [1] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, CK Luk, Bert Maher, Yunjie Pan, Christian Puhersch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Michael Suo, Phil Tillet, Eikan Wang, Xiaodong Wang, William Wen, Shunting Zhang, Xu Zhao, Keren Zhou, Richard Zou, Ajit Mathews, Gregory Chanan, Peng Wu, and Soumith Chintala. PyTorch 2: Faster Machine Learning Through Dy-

- dynamic Python Bytecode Transformation and Graph Compilation. ACM, 2024. [1](#)
- [2] Aram Davtyan, Sepehr Sameni, and Paolo Favaro. Efficient video prediction via sparsely conditioned flow matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. [2](#)
 - [3] Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable., 2022. [1](#)
 - [4] Marten Lienen and Stephan Günnemann. Torchode: A parallel ODE solver for PyTorch. In *The Symbiosis of Deep Learning and Differential Equations II, NeurIPS*, 2022. [1](#)
 - [5] Chen Liu and Tobias Ritschel. Neural Differential Appearance Equations. *ACM Trans. Graph.*, 43(6):256:1–256:17, 2024. [2](#)
 - [6] Mirela Ostrek and Justus Thies. Stable video portraits. In *European Conference on Computer Vision*, 2024. [2](#)
 - [7] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *arXiv preprint arXiv:1212.0402*, 2012. [2](#)
 - [8] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, Steven Liu, William Berman, Yiyi Xu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models, 2024. [1](#)