Clearer Frames, Anytime: Resolving Velocity Ambiguity in Video Frame Interpolation - Supplementary Materials

Zhihang Zhong[1,⋆] Gurunandan Krishnan[2] Xiao Sun[1] Yu Qiao[1]
Sizhuo Ma[2,†] Jian Wang[2,†]

[1] Shanghai Artificial Intelligence Laboratory
[2] Snap Inc.

1 Proof of Eq. (4)

Eq. (4) in the paper can be rigorously proven if an L2 loss is used,

$$\min_{\hat{I}_t} L = \mathbb{E}_{I_t \sim \mathcal{F}(I_0, I_1, t)}[(\hat{I}_t - I_t)^2]. \tag{1}$$

By setting the gradient to zero (this assumes during training, the neural network can reach the exact solution at this point),

$$\begin{aligned} \frac{\partial L}{\partial \hat{I}_t} &= 0 \\ \mathbb{E}_{I_t \sim \mathcal{F}(I_0, I_1, t)}[\frac{\partial}{\partial \hat{I}_t}(\hat{I}_t - I_t)^2] &= 0 \\ \mathbb{E}_{I_t \sim \mathcal{F}(I_0, I_1, t)}[2(\hat{I}_t - I_t)] &= 0 \\ \mathbb{E}_{I_t \sim \mathcal{F}(I_0, I_1, t)}[\hat{I}_t] - \mathbb{E}_{I_t \sim \mathcal{F}(I_0, I_1, t)}[I_t] &= 0 \\ \hat{I}_t &= \mathbb{E}_{I_t \sim \mathcal{F}(I_0, I_1, t)}[I_t] \end{aligned} \tag{2}$$

2 The Rationale for Solving Ambiguity

First and foremost, it is essential to clarify that **velocity ambiguity can solely exist and be resolved in the training phase, not in the inference phase**. The key idea behind our approach can be summarized as follows: While conventional VFI methods with time indexing rely on a one-to-many mapping, our distance indexing learns an approximate one-to-one mapping, which resolves the ambiguity during training. When the input-output relationship is one-to-many during training, the training process fluctuates among conflicting objectives, ultimately preventing convergence towards any specific optimization goal. In VFI, the evidence is the generation of blurry images in the inference phase. Once

⋆ First author, †Co-corresponding authors. This work was partially completed while Z. Zhong was a Snap Research intern (and a student at The University of Tokyo).

the ambiguity has been resolved using the new indexing method in the training phase, the model can produce significantly clearer results regardless of the inference strategy used.

Indeed, this one-to-many ambiguity in training is not unique to VFI, but for a wide range of machine learning problems. It is sometimes referred to as "mode averaging" in the community.[3] In some areas, researchers have come up with similar methods [5, 7].

A specific instantiation of this problem. Let us look at an example in text-to-speech (TTS). The same text can be paired with a variety of speeches, and direct training without addressing ambiguities can result in a **"blurred" voice** (a statistical average voice). To mitigate this, a common approach is to incorporate a speaker embedding vector or a style embedding vector (representing different gender, accents, speaking styles, etc.) during training, which helps reduce ambiguity. **During the inference phase, utilizing an average user embedding vector can yield high-quality speech output.** Furthermore, by manipulating the speaker embedding vector, effects such as altering the accent and pitch can also be achieved.

Here is a snippet from a high-impact paper Wang *et al.* [5] which came up with the style embedding in TTS:

> Many TTS models, including recent end-to-end systems, only learn an averaged prosodic distribution over their input data, generating less expressive speech – especially for long-form phrases. Furthermore, they often lack the ability to control the expression with which speech is synthesized.

Understanding this example can significantly help understand our paper, as there are many similarities between the two, *e.g.*, motivation, solution, and manipulation.

A minimal symbolic example to help understand better: Assuming we want to train a mapping function $\mathcal{F}$ from numbers to characters.
Training input-output pairs with ambiguity ($\mathcal{F}$ is optimized):

$$1 \xrightarrow{\mathcal{F}} a, 1 \xrightarrow{\mathcal{F}} b, 2 \xrightarrow{\mathcal{F}} a, 2 \xrightarrow{\mathcal{F}} b$$

$\mathcal{F}$ is optimized with some losses involving the input-output pairs above:

$$\min_{\mathcal{F}} \quad L(\mathcal{F}(1), a) + L(\mathcal{F}(1), b) + L(\mathcal{F}(2), a) + L(\mathcal{F}(2), b),$$

where L can be L1, L2 or any other kind of losses. Because the same input is paired with multiple different outputs, the model $\mathcal{F}$ is optimized to learn an average (or, generally, a mixture) of the conflicting outputs, which results in blur at inference.

[3] `https://www.cs.toronto.edu/~hinton/coursera/lecture13/lec13.pdf`

Inference phase ($\mathcal{F}$ is fixed):

$$1 \xrightarrow{\mathcal{F}} \{a, b\}?, 2 \xrightarrow{\mathcal{F}} \{a, b\}?$$

Training without ambiguity ($\mathcal{F}$ is optimized):

$$1 \xrightarrow{\mathcal{F}} a, 1 \xrightarrow{\mathcal{F}} a, 2 \xrightarrow{\mathcal{F}} b, 2 \xrightarrow{\mathcal{F}} b$$

In the input-output pairs above, each input value is paired with exactly one output value. Therefore, $\mathcal{F}$ is trained to learn a unique and deterministic mapping. Inference phase ($\mathcal{F}$ is fixed):

$$1 \xrightarrow{\mathcal{F}} a, 2 \xrightarrow{\mathcal{F}} b$$

Coming back to VFI. When time indexing is used, the same t value is paired to images where the objects are located at various locations due to the speed and directional ambiguities. When distance mapping is used, a single d value is paired to images where the objects are always at the same distance ratio, which allows the model to learn a more deterministic mapping for resolving the speed ambiguity.

It is important to note that fixing the ambiguity does not solve all the problems: At inference time, the "correct" (close to ground-truth) distance map is not available. In this work, we show that it is possible to provide uniform distance maps as inputs to generate a clear output video, which is not perfectly pixel-wise aligned with the ground truth. This is the reason why the proposed method does not achieve state-of-the-art in terms of PSNR and SSIM in Tab. 1 of the paper. However, it achieves sharper frames with higher perceptual quality, which is shown by the better LPIPS and NIQE.

We claim the "correct" distance map is hard to estimate accurately from merely two frames since there are a wide range of possible velocities. If considering more neighboring frames like Xu *et al.* [6] (more observation information), it is possible to estimate an accurate distance map for pixel-wise aligned interpolation, which we leave for future work.

Furthermore, manipulating distance maps corresponds to sampling other possible unseen velocities, *i.e.*, 2D manipulation of frame interpolation, similar to that mentioned TTS paper Wang *et al.* [5].

3 Additional Experiments and Analysis

Comparison of the fixed-time setting. While the benefits of our proposed disambiguation strategies are best demonstrated on arbitrary-time VFI models, they actually improve the performance of fixed-time models as well. Using RIFE [1] as a representative example, we extend our comparison to the fixed-time training paradigm, depicted in Fig. 1. The label [T] RIFE (Tri) refers to the model trained on the triplet dataset from Vimeo90K [8] employing time indexing. Conversely,

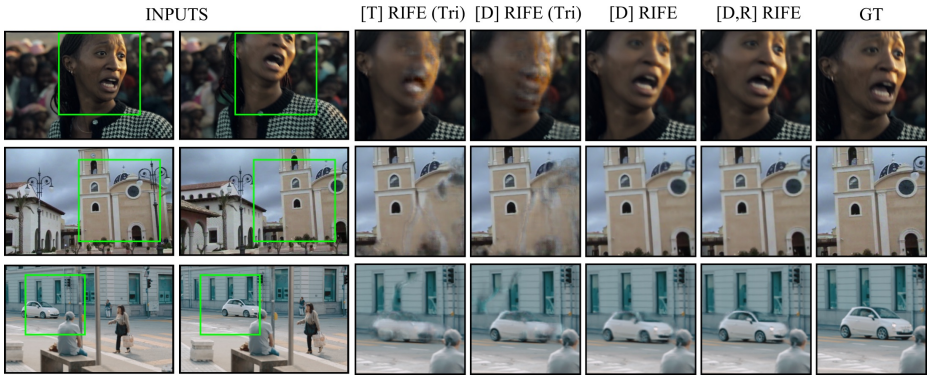

Fig. 1: Additional comparison of qualitative results. [T] RIFE (Tri) denotes RIFE [1] trained in a fixed time indexing paradigm (Vimeo90K triplet dataset [8]). [D] RIFE (Tri) denotes the model trained using distance indexing. All models use uniform maps.

Table 1: Comparison on Vimeo90K triplet dataset. $[T]$ denotes the method trained with traditional fixed time indexing paradigm. $[D]$ denotes the distance indexing paradigm. $[\cdot]_u$ denotes inference with uniform map as time indexes.

	RIFE [1]			IFRNet [2]			AMT-S [3]			EMA-VFI [9]		
	$[T]$	$[D]$	$[D]_u$	$[T]$	$[D]$	$[D]_u$	$[T]$	$[D]$	$[D]_u$	$[T]$	$[D]$	$[D]_u$
PSNR↑	35.61	**36.04**	35.18	35.80	**36.26**	35.14	35.97	**36.56**	35.21	36.50	**37.13**	36.21
SSIM↑	0.978	**0.979**	0.976	0.979	**0.981**	0.977	0.980	**0.982**	0.977	0.982	**0.983**	0.981
LPIPS↓	0.022	**0.022**	0.023	0.020	**0.019**	0.021	0.021	**0.020**	0.023	0.020	**0.019**	0.020
NIQE↓	5.249	5.225	**5.224**	5.256	5.245	**5.225**	5.308	5.293	**5.288**	5.372	5.343	**5.335**

$[D]$ RIFE (Tri) indicates training on the same triplet dataset but utilizing our distance indexing approach. Both $[D]$ RIFE and $[D, R]$ RIFE models are trained on the septuplet dataset, consistent with our earlier comparison. Despite being trained on varied datasets, it is evident that the arbitrary time model outperforms the fixed time model. However, the efficacy of distance indexing appears restrained within the fixed-time training paradigm. This limitation stems from the fact that deriving distance representation solely from the middle frame yields a sparse distribution, making it challenging for the network to grasp the nuances of distance. We delve deeper into the quantitative analysis of these findings in Tab. 1. Compared to training arbitrary time models on the septuplet dataset, the advantages of distance indexing become notably decreased when training fixed time models on the triplet dataset.

Comparison of using perceptual loss. In addition to training with traditional pixel losses based on L1 and L2 losses, we present the results of employing the

Table 2: Comparison on the septuplet of Vimeo90K [8] using LPIPS loss [10]. We use RIFE [1] as a representative example.

	$[T]$	$[D]_u$	$[D, R]_u$
PSNR ↑	**27.19**	26.71	26.72
SSIM ↑	**0.898**	0.889	0.890
LPIPS ↓	**0.061**	0.065	0.064
NIQE ↓	6.307	5.901	**5.837**

Table 3: Comparison with $[D_{x,y}]$ scheme on Vimeo90K septuplet dataset. $[D_{x,y}]$ denotes training with two channel distance map with $\boldsymbol{x}$ and $\boldsymbol{y}$ directions. $[\cdot]_u$ denotes inference with uniform maps.

	$[T]$	$[D]_u$	$[D, R]_u$	$[D_{x,y}]_u$	$[D_{x,y}, R]_u$
LPIPS↓	0.105	0.092	**0.086**	0.091	0.087
NIQE↓	6.663	6.344	**6.220**	6.296	**6.220**

more recent LPIPS loss [10] with a VGG backbone [4], as shown in Tab. 2. The non-reference perceptual quality metric, NIQE, shows notable improvement across all variants. The results also consistently demonstrate the effectiveness of our strategies in resolving velocity ambiguity. Besides, due to the direct optimization of LPIPS loss, the assumption of constant speed in uniform maps affects the performance for this metric. This is why in the test results, $[T]$ has a lower LPIPS, while $[D]_u$ and $[D, R]_u$ are slightly higher.

Two channel scheme. Comparison results with the two channel scheme, *i.e.*, D_t with $\boldsymbol{x}$ and $\boldsymbol{y}$ directions, are shown in Tab. 3. The observations are as follows: (1) $[D_{x,y}]_u$ outperforms $[D]_u$, which makes sense since it accounts for both speed and direction. (2) $[D, R]_u$ performs better than $[D_{x,y}]_u$. Our speculation is that $[R]$ benefits not only from addressing the directional ambiguity but also from reducing the prediction difficulty via divide-and-conquer. (3) $[D_{x,y}, R]_u$ does not exceed $[D, R]_u$, showing that the iterative formulation is sufficient to resolve the directional ambiguity. $[D_{x,y}]_u$ has the potential to learn different trajectories but cannot realize that potential since the trajectory distribution within a short time is not sufficiently diverse. However, $[D_{x,y}]_u$ only involves increasing the input channels and does not need to run iteratively. Thus, $[D_{x,y}]_u$ is a better choice for fast/lightweight frame interpolation.

Why did $[R]$ *fail on AMT-S?* As compared to the speed ambiguity, the directional ambiguity only has a minor impact to the interpolation quality due to the short time span between frames. Merely employing iterative reference-based

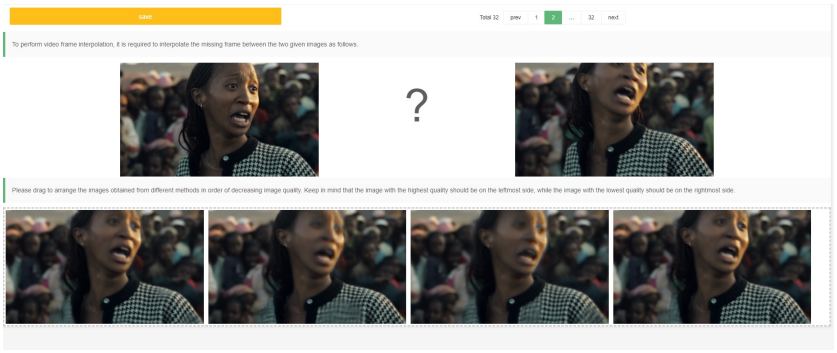

Fig. 2: User interface of user study.

estimation without tackling speed ambiguity $[T, R]$ can result in the accumulation of inaccuracies. This phenomenon is especially evident in AMT-S, which we attribute to its scaled lookup operation of bidirectional 4D correlation volumes. Cumulative errors are exacerbated in inaccurate iterative lookup operations.

Why NIQE performs better with uniform maps? Uniform maps tend to yield better results due to smoothing. Nonuniform maps consist of unavoidable inaccuracies from flow estimation, which may introduce unnatural details, resulting in worse NIQE scores but are not noticeable to human perception.

4 Costs of Proposed Strategies

Distance indexing. Transitioning from time indexing ($[T]$) to distance indexing ($[D]$) does not introduce extra computational costs during the inference phase, yet significantly enhancing image quality. In the training phase, the primary requirement is a one-time offline computation of distance maps for image triplets.

Iterative reference-based estimation. Given that the computational overhead of merely expanding the input channel, while keeping the rest of the structure unchanged, is negligible, the computational burden during the training phase remains equivalent to that of the $[D]$ model. Regarding inference, the total consumption is equal to the number of iterations $\times$ the consumption of the $[D]$ model. We would like to highlight that this iterative strategy is optional: Users can adopt this strategy at will when optimal interpolation results are demanded and the computational budget allows.

5 User Study UI

As shown in Fig. 2, we initially presented users with the input starting and ending frames. Subsequently, the results from each model's four distinct variants were displayed anonymously in a sequence, with the order shuffled for each

presentation. Users were tasked with reordering the images by dragging them, placing them from left to right based on their perceived quality, *i.e.*, the best image on the extreme left and the least preferred on the far right.

6 Demo

We have included a video demo (available in supplementary materials named "supp.mp4") to intuitively showcase the enhanced quality achieved through our strategies. The video further illustrates the idea of manipulating object interpolations and provides a guide on using the related web application.

References

1. Huang, Z., Zhang, T., Heng, W., Shi, B., Zhou, S.: Real-time intermediate flow estimation for video frame interpolation. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIV. pp. 624–642. Springer (2022)
2. Kong, L., Jiang, B., Luo, D., Chu, W., Huang, X., Tai, Y., Wang, C., Yang, J.: Ifrnet: Intermediate feature refine network for efficient frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1969–1978 (2022)
3. Li, Z., Zhu, Z.L., Han, L.H., Hou, Q., Guo, C.L., Cheng, M.M.: Amt: All-pairs multi-field transforms for efficient frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9801–9810 (2023)
4. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
5. Wang, Y., Stanton, D., Zhang, Y., Ryan, R.S., Battenberg, E., Shor, J., Xiao, Y., Jia, Y., Ren, F., Saurous, R.A.: Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In: International conference on machine learning. pp. 5180–5189. PMLR (2018)
6. Xu, X., Siyao, L., Sun, W., Yin, Q., Yang, M.H.: Quadratic video interpolation. Advances in Neural Information Processing Systems **32** (2019)
7. Xu, Y., Tan, H., Luan, F., Bi, S., Wang, P., Li, J., Shi, Z., Sunkavalli, K., Wetzstein, G., Xu, Z., et al.: Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. arXiv preprint arXiv:2311.09217 (2023)
8. Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video enhancement with task-oriented flow. International Journal of Computer Vision **127**, 1106–1125 (2019)
9. Zhang, G., Zhu, Y., Wang, H., Chen, Y., Wu, G., Wang, L.: 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. pp. 5682–5692 (2023)
10. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)