AnimateDiff-Lightning: Cross-Model Diffusion Distillation

Shanchuan Lin Xiao Yang
ByteDance Inc.
{peterlin, yangxiao.0}@bytedance.com

Abstract

We present AnimateDiff-Lightning for lightning-fast video generation. Our model uses progressive adversarial diffusion distillation to achieve new state-of-the-art in few-step video generation. We discuss our modifications to adapt it for the video modality. Furthermore, we propose to simultaneously distill the probability flow of multiple base diffusion models, resulting in a single distilled motion module with broader style compatibility. We are pleased to release our distilled AnimateDiff-Lightning model for the community's use.

Model: https://huggingface.co/ByteDance/AnimateDiff-Lightning

1. Introduction

Video generative models are gaining great attention lately. Text-to-video models [2–4, 6, 8, 30, 36, 44] allow the creation of videos straight from ideation; image-to-video models [2, 4, 6, 36] enable more fine-grained control over content and composition; video-to-video models [4, 6] can convert existing videos to different styles, such as anime or cartoon. The advancement in video generation has enabled brand-new creative possibilities.

Among all methods, AnimateDiff [6] is one of the most popular video generation models. It takes a frozen image generation model and injects learnable temporal motion modules into the network. This allows the model to inherit the image priors and learn to produce temporally coherent frames from limited video datasets. Since the im-

age model's architecture and weights are unchanged, it can be swapped with a wide range of stylized models post-training to create amazing anime and cartoon videos, *etc*. Additionally, AnimateDiff is compatible with image control modules, such as ControlNet [42], T2I-Adapter [22], IP-Adapter [40], *etc*., which further enhance its versatility.

However, speed is one of the main hurdles preventing video generation models from wider adoption. State-of-the-art generative models are slow and computationally expansive due to the iterative diffusion process. This issue is further worsened in video generation. For example, many video stylization pipelines using AnimateDiff with ControlNet and a stylized image model can take up to ten minutes to process a ten-second video. Making the generation faster while retaining its quality is the main focus of this work.

Diffusion distillation [11, 13, 17, 18, 20, 21, 28, 29, 31, 32, 35, 41, 43] has been more widely researched in image generation. Recently, progressive adversarial diffusion distillation [13] has achieved state-of-the-art results in few-step image generation. In this paper, we apply it to video models for the first time, demonstrating the applicability and superiority of this method on the video modality. We will discuss our designs and changes made specifically for video model distillation.

In addition, we propose to simultaneously distill the probability flow of multiple base diffusion models. Specifically, we take special consideration into the fact that AnimateDiff is widely used with different stylized base models. However, all existing methods perform distillation only on the default base model, and can only hope that the distilled motion module will still work after swapping onto a new base. In practice, we find the quality degrades as the inference step reduces. Therefore, we propose to explicitly and simultaneously distill a shared motion module on different base models. We find this approach not only improves quality on the selected base models, but also on unseen base models.

Our proposed AnimateDiff-Lightning can generate better quality videos in fewer inference steps, out-competing the prior video distillation method AnimateLCM [35]. We release our distilled AnimateDiff-Lightning model for the community's use.

2. Background

2.1. Diffusion Model

Diffusion models [9, 33] are behind most state-of-the-art video generation methods. The generation involves a probability flow [16, 17, 33] that gradually transports samples x_t from the noise distribution $t = T$ to the data distribution $t = 0$. A neural network f is learned to predict the gradient at any location of this flow. Because the flow is curved and complex, the generation must only take a small step along the gradient at a time, repeatedly invoking expansive neural network evaluations. Diffusion distillation trains the neural network to directly predict the next flow location farther ahead, allowing traversing the flow with bigger strides and fewer steps.

2.2. Progressive Adversarial Diffusion Distillation

Progressive adversarial diffusion distillation [13] proposes to combine progressive distillation [28] and adversarial loss [5]. Specifically, progressive distillation [28] trains a student network to directly predict the next flow location x_{t-ns} from the current flow location x_t as if the teacher network has stepped through n steps of stride s. After the student converges, it is used as the teacher and the process repeats itself for further distillation:

$$x_{t-ns} = \mathbf{EulerSolver}(f_{\text{teacher}}, x_t, t, c, n, s) \quad (1)$$

$$\hat{x}_{t-ns} = \mathbf{EulerSolver}(f_{\text{student}}, x_t, t, c, 1, ns) \quad (2)$$

$$\mathcal{L}_{\text{mse}} = \|\hat{x}_{t-ns} - x_{t-ns}\|_2^2 \quad (3)$$

However, theoretical analysis [13] has shown that exact matching with mean squared error (MSE) as in Equation (3) is impossible due to reduced model capacity, so adversarial loss is introduced to trade-off between quality and mode coverage. The method proposes to first distill with discriminator D conditioned on x_t and caption c to enforce flow trajectory preservation:

$$p = D(x_t, x_{t-ns}, t, t-ns, c) \quad (4)$$

$$\hat{p} = D(x_t, \hat{x}_{t-ns}, t, t-ns, c) \quad (5)$$

Then, distill with discriminator D' without the condition on x_t to relax the trajectory requirement to improve quality:

$$p = D'(x_{t-ns}, t-ns, c) \quad (6)$$

$$\hat{p} = D'(\hat{x}_{t-ns}, t-ns, c) \quad (7)$$

The distillation trains the diffusion model and the discriminator with non-saturated adversarial loss [5] in alternating iterations:

$$\mathcal{L}_D = -\log(p) - \log(1-\hat{p}) \quad (8)$$

$$\mathcal{L}_G = -\log(\hat{p}) \quad (9)$$

SDXL-Lightning [13] achieves new state-of-the-art in one-step/few-step text-to-image generation with this distillation method. Our work is the first to apply this method in video diffusion distillation, demonstrating the applicability and superiority of the method in other modalities.

2.3. Other Diffusion Distillation Methods

Diffusion distillation is mostly studied in image generation. Most notably, Latent Consistency Model (LCM) [20,

21] applies consistency distillation [32] for latent image diffusion models; InstaFlow [18] uses a technique called rectified flow (RF) [17] to gradually make the flow straighter as a way to reduce sampling steps; SDXL-Turbo [29] uses adversarial loss with score distillation sampling (SDS) [24] to push generation down to one step. SDXL-Lightning [13] is the latest research in distillation and achieves even better quality compared to previous methods with progressive adversarial distillation.

Research on video diffusion distillation is very scarce. AnimateLCM [35] is the only work on video diffusion distillation so far to the best of our knowledge. It follows LCM [20, 21] to apply consistency distillation [32] on AnimateDiff. AnimateLCM can generate great quality videos with eight inference steps but starts to show artifacts with four inference steps, and the results are blurry under four inference steps.

2.4. Distillation as Pluggable Modules

LCM [21], AnimateLCM [35], and SDXL-Lightning [13] have explored training the distillation as a pluggable module. The module contains additional parameters on top of the frozen base model, allowing the module to be transplanted onto other stylized base models post-training.

However, the distillation module is only trained on the default base model and the whole approach depends on the assumption that other stylized base models have similar weights. Empirically, we find the quality degrades as the inference step reduces on unseen base models.

In this paper, we explore explicitly and simultaneously distilling the distillation module on multiple base models for the first time. This provides a quality guarantee on the selected base models. We also find it improves compatibility on unseen base models.

3. Method

We propose to train a shared distilled motion module on multiple base models simultaneously for AnimateDiff [6]. The resulting motion module has better few-step inference compatibility with different base models.

3.1. Model and Data Preparation

Besides the default Stable Diffusion (SD) v1.5 base model [26], we select multiple additional target base models based on their popularity. For realistic style, we select RealisticVision v5.1 [56] and epiCRealism [49]. For anime style, we select ToonYou Beta 6 [58], IMP v1.0 [51], and Counterfeit v3.0 [46].

The existing video dataset WebVid-10M [1] only contains realistic stock video footage. The samples are especially out-of-distribution when distilling the anime models. Therefore, we apply AnimateDiff on all the selected base models to mass-generate data samples. Specifically, we generate video clips using the prompts from WebVid-10M [1]. We use DPM-Solver++ [19] with 32 steps and a classifier-free guidance (CFG) scale of 7.5 without negative prompts. All the clips are 16 frames and 512×512 resolution. In total, we have generated 1.75 million clips.

3.2. Cross-Model Distillation

The AnimateDiff model F_i is composed of the frozen image base model f_i and the shared motion module m, where i denotes the index of the specific base model.

$$F_i := f_i \circ m \tag{10}$$

At distillation, we only update the weights of the motion module and keep the weights of the image base model unchanged. We load different image base model f_i on different GPU ranks and initialize the motion module m with the same AnimateDiff v2 checkpoint [6]. The specific assignments are shown in Table 1.

This design allows the motion module to be simultaneously distilled on multiple base models. Spreading different base models across GPUs eliminates the need for constant swapping of the base models on each GPU. We modify the PyTorch Distributed Data Parallel (DDP) framework [23] to prevent synchronization of the frozen image base model from erasing our model assignments. After the modification, the gradients are automatically accumulated using the existing distributed training mechanism to ensure optimization toward accurate distillation on all base models.

We also assign different distillation datasets according to the image base model. For distilling the Stable Diffusion base model, we use the WebVid-10M dataset [1]. For distilling each realistic or anime model, we pool together all the generated data of its kind to improve diversity. We also employ random horizontal flips to double the sample count.

Rank	Base Model	Dataset
0	Stable Diffusion v1.5 [26]	WebVid-10M [1]
1	Stable Diffusion v1.5 [26]	
2	RealisticVision v5.1 [56]	Generated Realistic
3	epiCRealism [49]	
4	ToonYou Beta 6 [58]	Generated Anime
5	ToonYou Beta 6 [58]	
6	IMP v1.0 [51]	
7	Counterfeit v3.0 [46]	

Table 1. Model and dataset assignments across 8 GPU ranks in a single machine. The same configuration is replicated to additional machines.

3.3. Flow-Conditional Video Discriminator

Progressive adversarial diffusion distillation [13] proposes to use discriminator D to ensure that the student prediction of x_{t-ns} from x_t given caption c is sharp and flow-preserving. Since our distillation now involves multiple flows of different base models, we must extend the discriminator to be flow-conditional. Specifically, we provide the corresponding base model index i to the discriminator. This way the discriminator can learn and critique separate flow trajectories for each base model:

$$\begin{aligned} &D(x_t, x_{t-ns}, t, t-ns, c, i) \\ &\quad := \sigma\Big(\text{head}\Big(d(x_{t-ns}, t-ns, c, i), d(x_t, t, c, i)\Big)\Big) \end{aligned} \tag{11}$$

We follow prior works [13, 15] to take the diffusion UNet [27] encoder and midblock as the discriminator backbone d. In our case, we use the AnimateDiff architecture [6], which consists of the image base model initialized with SD v1.5 weights [26] and the motion module initialized with AnimateDiff v2 weights [6]. We include flow condition i as a new learnable embedding and add it to the time embedding. The shared backbone processes $d(x_{t-ns}, t-ns, c, i)$ and $d(x_t, t, c, i)$ independently. The resulting midblock features are concatenated along the channel dimension before passing to a prediction head. The prediction head consists of blocks of 3D convolution with a kernel size of 4 and a stride of 2, group normalization [37], and SiLU activation [7, 25] to further reduce the dimension to a single value. Finally, the sigmoid function $\sigma(\cdot)$ clamps the value to $[0, 1]$ range, denoting the probability of the input x_{t-ns} being generated from the teacher as opposed to the student. The entire discriminator, including the backbone, is trained.

Progressive adversarial diffusion distillation [13] also proposes to further finetune the model without condition on x_t at each stage to relax the flow trajectory preservation requirement and further improve the quality. But note that despite the flow trajectory preservation is relaxed, we still must enforce the student prediction to be within the distribution of the target flow. Therefore, we also modify this discriminator D' to be conditional on flow i:

$$\begin{aligned} &D'(x_{t-ns}, t-ns, c, i) \\ &\quad := \sigma\Big(\text{head}\Big(d(x_{t-ns}, t-ns, c, i)\Big)\Big) \end{aligned} \tag{12}$$

3.4. Distillation Procedure

We progressively distill the model in the following step count order: $128 \rightarrow 32 \rightarrow 8 \rightarrow 4 \rightarrow 2$. We use mean squared error (MSE) and apply classifier-free guidance (CFG) on $128 \rightarrow 32$ distillation. The CFG scale is set to 7.5 and no negative prompts. We use adversarial loss for the rest of the stages. Note that our data generation uses DPM-Solver++ [19] for 32 steps. Since DPM-Solver++ produces better quality than Euler, we still decide to start the distillation from 128 steps for extra quality.

The distillation is performed on 64 A100 GPUs. Each GPU can only process a batch size of 1 due to the memory constraint, so we apply a gradient accumulation of 4 to achieve a total batch size of 256. Other hyperparameters, such as learning rate, *etc*., follow SDXL-Lightning [13] exactly. We adopt the linear schedule [9] as used in the original AnimateDiff but use pure noise at the last timestep as model input during training following [13] to ensure zero terminal SNR [12].

Unlike SDXL-Lightning [13], we cannot switch to x_0-prediction while keeping the base model frozen for one-step generation, so we train the model in ϵ-prediction.

Compared to AnimateLCM [35], which first distills the image base model as a LoRA module [10] on image datasets and then distills the video motion module on limited video datasets to combat data scarcity, our method distills the whole AnimateDiff model as a whole. Furthermore, we find the distillation can be trained on the motion module alone for satisfactory quality and there is no need for an additional LoRA module on the image base model.

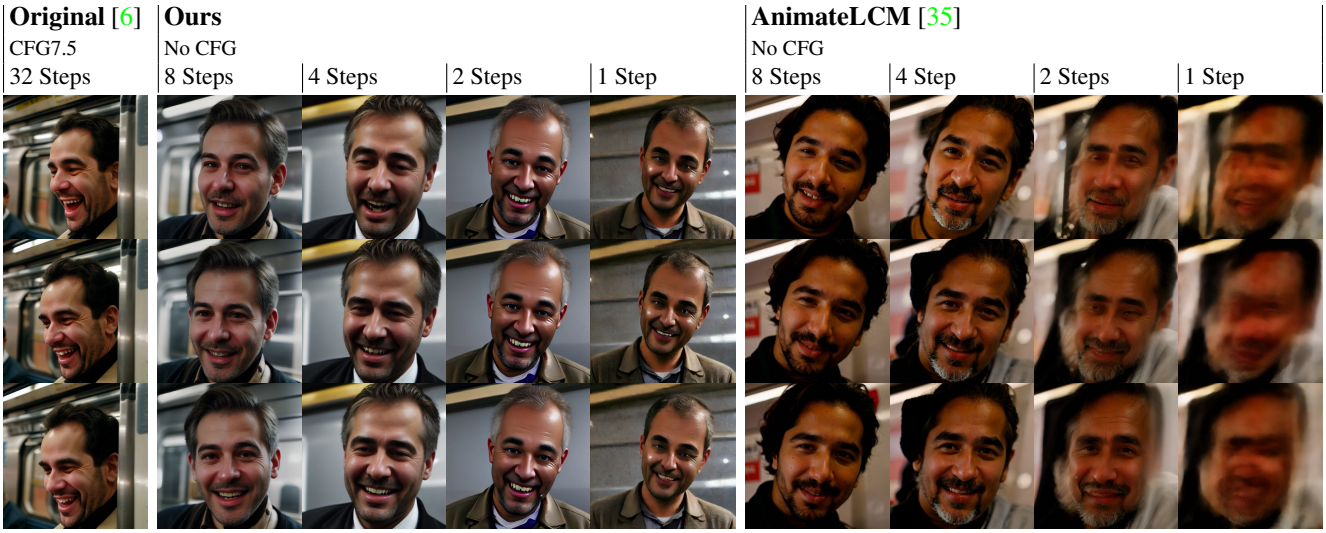
4. Evaluation

4.1. Qualitative Evaluation

Figure 1 shows qualitative comparison of our model to the original AnimateDiff [6] and AnimateLCM [35]. Our method achieves better quality with 1-step, 2-step, and 4-step inference compared to AnimateLCM. The difference is particularly pronounced when using 1-step and 2-step inference as AnimateLCM fails to generate sharp details. Additionally, our method using cross-model distillation can better retain the original style of the base model. AnimateLCM sometimes over-exposes and differs from the base model's style and tone even when using 8-step inference.

Figure 1f shows the results of our model when applied to an unseen base model: Mistoon Anime v1.0 [54]. The style gradually deviates from the original style as the inference step reduces, but note that our model still generates results closer to the original compared to AnimateLCM in terms of the overall anime style, clothing, and hair color of the characters. More analysis on the effect of cross-model distillation is provided in Section 5.1. More analysis on unseen models is provided in Section 5.2

The 1-step model produces heavy noise artifacts. This is likely due to the numerical instability of the epsilon formulation, which is also encountered by SDXL-Lightning [13]. For the 2-step model, we notice that it produces more pronounced brightness flickers. Note that the flickers have existed since the original AnimateDiff model. We find the 4-step model strikes the balance between quality and speed.

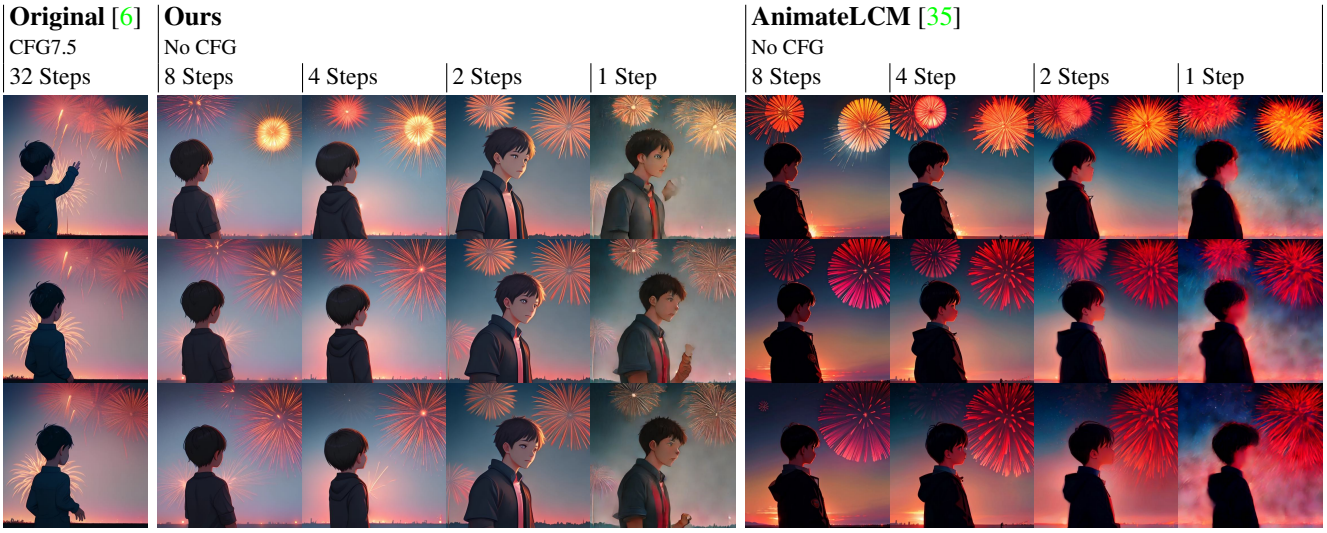


(a) epiCRealism [49]: A close-up of a man talking and laughing on New York subway. (Our method generates sharper details in 2 steps and 1 step.)

(b) RealisticVision v5.1 [56]: A man holding a black umbrella running in a rainy day. (Our method matches the original tone and style better.)

(c) epiCRealism [49]: Entering a big castle. (Our method generates sharper details in 2 steps and 1 step.)

Figure 1. Qualitative Comparison. We only show the first, middle, and last frames of the generated video clips in each column. Our model generates better results using 1-step, 2-step, and 4-step inference. Additionally, our model can better retain the style of the original model. This page focuses on realistic style generation. Please see the next page for anime-style generation.

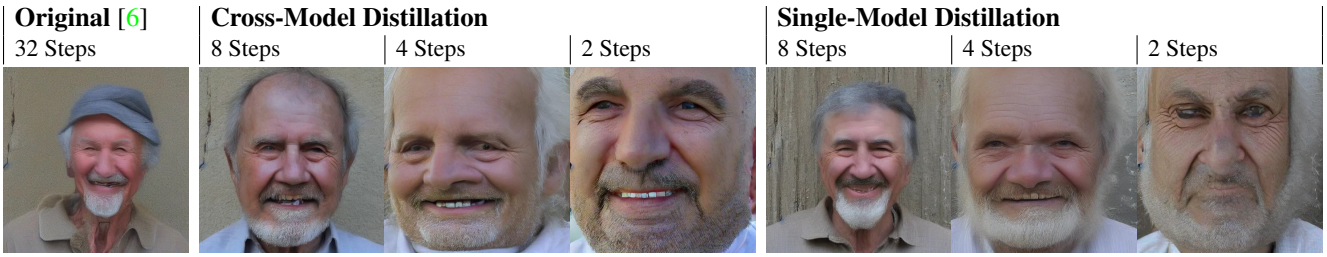


(d) IMP v1.0 [51]: A boy looking at the sky, firework in the background. (Our method matches the original tone and style better.)

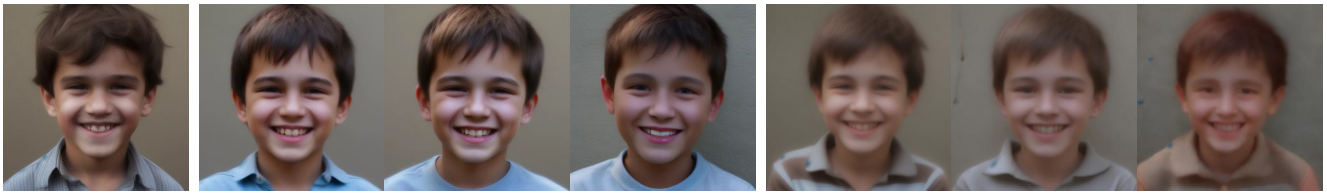
(e) ToonYou Beta 6 [58]: A girl smiling. (Our method matches the original tone and style better.)

(f) Mistoon Anime [54]: A couple dancing at the beach. (On an unseen base model, our method matches the original style, clothing, and hair color better.)

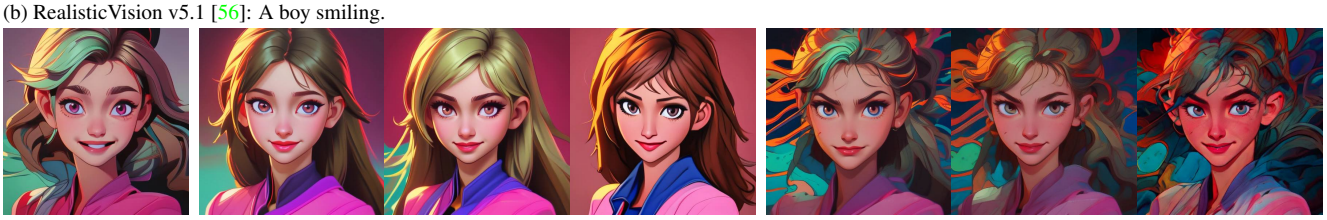
Figure 1. Qualitative Comparison. Continuing from the last page, we show an anime-style generation comparison on this page. We also try to apply our model on an unseen base model: Mistoon Anime [54] in Fig. 1f. Though there is style degradation as the inference step reduces, our model produces more similar results compared to the original in terms of overall anime style, clothing, and hair color of the characters.



(a) Stable Diffusion v1.5 [26]: An old man smiling.



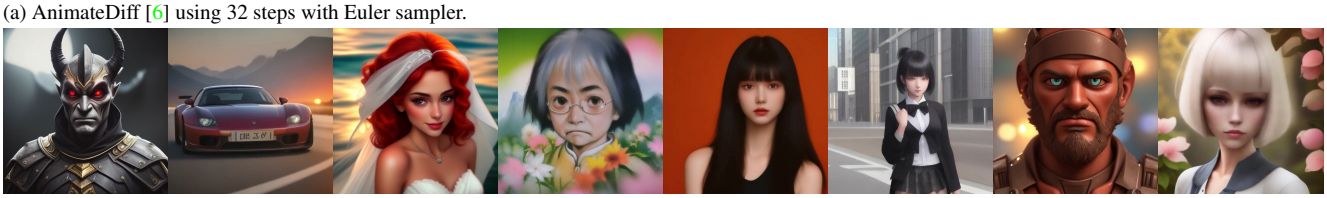
(b) RealisticVision v5.1 [56]: A boy smiling.



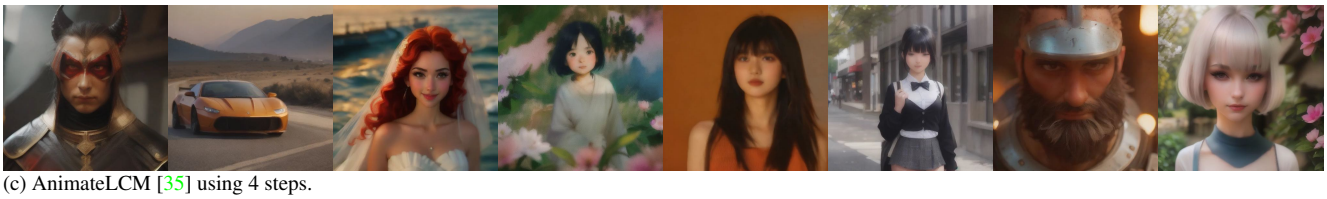
(c) ToonYou Beta 6 [58]: A girl smiling.

Figure 2. Comparison between cross-model and single-model distillation. Single-model distillation is trained only on SD v1.5 [26] base model with the WebVid-10M [1] dataset. Single-model distillation fails to retain quality on other base models. We show the first frame of the generated video clips.

(a) AnimateDiff [6] using 32 steps with Euler sampler.



(b) Our method using 4 steps.

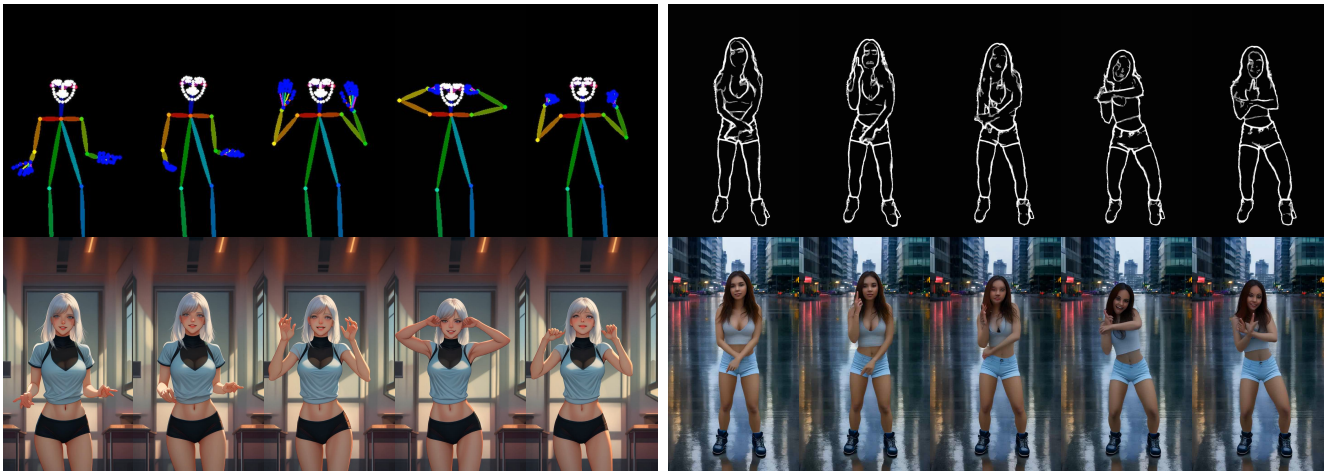


(c) AnimateLCM [35] using 4 steps.

Figure 3. Distillation results on unseen base models. All the image base models here are unseen during the distillation of our model and the AnimateLCM model [35]. Our results are better in detail and are closer to the original styles. We use different prompts that best match the image base models' specialty, but the same prompt and seed are used across model comparisons. We show the first frame of the generated video clips.

Figure 4. Our model is compatible with Motion LoRA modules [6] for fine-grained motion control. Here is our 4-step model on ToonYou [58] with prompt: "A girl smiling". The first row is the starting frame and the second row is the final frame.

Figure 5. Text-to-video generation of different aspect ratios. Examples shown here are 2-step and 4-step models generating 1:2, 2:3, 3:2, and 2:1 aspect ratios. We show a random frame from the generated video clips.



(a) 4 Steps, IMP v1.0 [51], DWPose [39]

(b) 2 Steps, epiCRealism [49], HED [38], RobustVideoMatting [14]

Figure 6. Video-to-video generation with ControlNet [42]. The example videos are generated in 576×1024 resolution directly using our model with ControlNet [42]. More sophisticated pipelines, such as using super-resolution, can further enhance the quality.

4.2. Quantitative Evaluation

Method	Steps	FVD ↓			
		RV [56]	TY [58]	DS [47]	DV [48]
AnimateLCM	1	1423.18	1825.24	1393.10	1652.32
	2	1041.61	917.61	1034.19	1045.49
	4	1171.54	784.81	1175.06	1097.66
	8	1300.41	804.21	1253.43	1115.95
Ours	1	1135.43	1037.85	974.75	1501.34
	2	1024.13	801.04	918.74	1351.06
	4	1010.30	708.55	908.01	1175.29
	8	1058.58	690.65	865.29	979.94

Table 2. FVD computed against original AnimateDiff on different image base models. RV: RealisticVision, TY: ToonYou, DS: DreamShaper, DV: DynaVision.

Table 2 shows quantatitive comparison. First, we randomly select 100 prompts from the WebVid-10M dataset [1]. Then, we generate the clips using four different image base models. We select RealisticVision [56] and ToonYou [58] as seen realistic and anime style models, and select DreamShaper [47] and DynaVision [48] as unseen realistic and anime style models. Each prompt uses a random seed but the same seed is used across models on the same prompt. Finally, we compute FVD [34] against the original AnimateDiff results generated using 32 Euler steps and CFG 7.5 without negative prompts. Both ours and AnimateLCM [35] do not use CFG. The metrics show that our models have better FVD compared to AnimateLCM and therefore produce results closer to the original AnimateDiff.

5. Ablation

5.1. Effects of Cross-Model Distillation

We conduct a comparison experiment to distill a model only using Stable Diffusion v1.5 [26] as the image base model on the WebVid-10M [1] dataset. This corresponds to the regular single-model distillation paradigm.

Figure 2 shows that single-model distillation can only keep the best quality on the default SD [26] base model. The quality degrades after switching to RealisticVision [56] which has a similar realistic style. The quality significantly degrades after switching to ToonYou [58] which has a drastically different anime style.

5.2. Effects on Unseen Base Models

We test our model on a wide variety of popular image base models. These base models are unseen during the distillation process. Figure 3 shows that our distilled motion module can generalize well to other unseen base models. Furthermore, our distilled model produces results with sharper details and closer styles to the original model compared to AnimateLCM [35].

5.3. Compatibility with Motion LoRAs

Figure 4 shows that our model is compatible with Motion LoRAs [6]. We have tested Motion LoRAs on all our models and have found that they work in all step settings. We apply Motion LoRAs with a strength of 0.8 to avoid watermarks, an issue Motion LoRAs introduce. We find Motion LoRAs enable fine-grained control of the camera motion and they greatly enhance the amount of motion in the generated videos.

5.4. Support for Different Aspect-Ratios

Figures 5 and 6 show that our model retains the ability to generate videos of different aspect-ratios on both text-to-video and video-to-video scenarios despite the distillation is performed only in square aspect ratio. However, we find that as the aspect ratio deviates more from the square, there is a higher probability of generating bad cases. The distillation training can be done in multiple aspect ratios. We leave this to future improvements.

5.5. Video-to-Video Generation with ControlNet

One of AnimateDiff's most popular uses is video-to-video stylization. Given a source video, ControlNet [42] is applied to extract human movement, and then AnimateDiff is used to generate the movement using different styles.

Figure 6 shows that our model is compatible with ControlNet [42]. Here we only apply the basic setting, but a more sophisticated pipeline, such as using super-resolution and background replacement, can be additionally added. To generate longer videos, the popular approach is context overlapping, which overlaps the 16-frame context window with previously generated clips. We have tested that our models support generating longer videos with context overlapping.

6. Conclusion

We have presented AnimateDiff-Lightning, a lightning-fast video generation model. In this paper, we have shown that progressive adversarial diffusion distillation can be applied in the video modality. Our model achieves new state-of-the-art in few-step video generation. Additionally, we have proposed cross-model diffusion distillation to further improve the distillation module's ability to generalize to different stylized base models. We apply the cross-model distillation technique on AnimateDiff because it is most widely used with different image base models. However, this technique can be generalized to create more universal distillation pluggable modules for all modalities. Lastly, we release our distilled AnimateDiff-Lightning models with the hope of facilitating advancements in generative AI.

References

[1] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1708–1718, 2021. 3, 7, 9

[2] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets, 2023. 1

[3] A. Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22563–22575, 2023. 1

[4] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7312–7322, 2023. 1

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63:139 – 144, 2014. 2

[6] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 3, 4, 5, 6, 7, 8, 9

[7] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2023. 4

[8] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models, 2022. 1

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 2, 4

[10] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 4

[11] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[12] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5404–5411, January 2024. 4

[13] Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation, 2024. 2, 3, 4

[14] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3132–3141, 2021. 8

[15] Shanchuan Lin and Xiao Yang. Diffusion model with perceptual loss, 2024. 4

[16] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[17] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. 2, 3

[18] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and qiang liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 3

[19] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models, 2023. 3, 4

[20] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference, 2023. 2, 3

[21] Simian Luo, Yiqin Tan, Suraj Patil, Daniel Gu, Patrick von Platen, Apolinário Passos, Longbo Huang, Jian Li, and Hang Zhao. Lcm-lora: A universal stable-diffusion acceleration module, 2023. 2, 3

[22] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models, 2023. 2

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. 3

[24] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2023. 3

[25] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2017. 4

[26] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis

with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2021. 3, 4, 7, 9

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597, 2015. 4

[28] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. 2

[29] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation, 2023. 2, 3

[30] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*, 2023. 1

[31] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[32] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, 2023. 2, 3

[33] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 2

[34] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *ArXiv*, abs/1812.01717, 2018. 9

[35] Fu-Yun Wang, Zhaoyang Huang, Xiaoyu Shi, Weikang Bian, Guanglu Song, Yu Liu, and Hongsheng Li. Animatelcm: Accelerating the animation of personalized diffusion models and adapters with decoupled consistency learning, 2024. 2, 3, 4, 5, 6, 7, 9

[36] Weimin Wang, Jiawei Liu, Zhijie Lin, Jiangqiao Yan, Shuo Chen, Chetwin Low, Tuyen Hoang, Jie Wu, Jun Hao Liew, Hanshu Yan, Daquan Zhou, and Jiashi Feng. Magicvideo-v2: Multi-stage high-aesthetic video generation, 2024. 1

[37] Yuxin Wu and Kaiming He. Group normalization. *International Journal of Computer Vision*, 128:742 – 755, 2018. 4

[38] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. *International Journal of Computer Vision*, 125:3 – 18, 2015. 8

[39] Zhendong Yang, Ailing Zeng, Chun Yuan, and Yu Li. Effective whole-body pose estimation with two-stages distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4210–4220, 2023. 8

[40] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models, 2023. 2

[41] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation, 2023. 2

[42] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3813–3824, 2023. 2, 8, 9

[43] Jianbin Zheng, Minghui Hu, Zhongyi Fan, Chaoyue Wang, Changxing Ding, Dacheng Tao, and Tat-Jen Cham. Trajectory consistency distillation, 2024. 2

[44] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models, 2023. 1

[45] AbsoluteReality v1.8.1. https://civitai.com/models/81458. 7

[46] Counterfeit v3.0. https://civitai.com/models/4468. 3

[47] DreamShaper v8. https://civitai.com/models/4384. 7, 9

[48] DynaVision v2. https://civitai.com/models/75549. 7, 9

[49] epiCRealism. https://civitai.com/models/25694. 3, 5, 8

[50] Exquisite Details Art. https://civitai.com/models/118495. 7

[51] IMP v1.0. https://civitai.com/models/56680. 3, 6, 8

[52] MajicMix Realistic v7. https://civitai.com/models/43331. 7

[53] MajicMix Reverie v1. https://civitai.com/models/65055. 7

[54] Mistoon Anime v1.0. https://civitai.com/models/24149. 4, 6

[55] RCNZ Cartoon 3d v2. https://civitai.com/models/66347. 7

[56] Realistic Vision v5.1. https://civitai.com/models/4201. 3, 5, 7, 9

[57] ReV Animated v1.2.2. https://civitai.com/models/7371. 7

[58] ToonYou Beta 6. https://civitai.com/models/30240. 3, 6, 7, 8, 9