Phased DMD: Few-step Distribution Matching Distillation via Score Matching within Subintervals

Supplementary Material

A. Related Work

Our work is situated within the framework of Variational Score Distillation (VSD) [43]. VSD involves three components: a trainable generator, a fake score estimator, and a pretrained teacher score estimator. The generator is optimized to produce a distribution that approximates the real data distribution. Concurrently, the fake score estimator learns to estimate the score of the generator's output distribution. The update direction for the generator is then determined by the discrepancy between the teacher's score (for the real distribution) and the fake score estimator's score.

Similar to GANs, the VSD framework is adversarial. The fake score estimator must be precisely optimized to learn the score of the current generated distribution. This accurate estimation is crucial, as it combines with the fixed teacher model (which provides the score for the real data) to produce a correct guidance signal for the generator. This principle explains why DMD2 [47] operates successfully without external real data, in contrast to its predecessor DMD [48].

A key advantage of VSD over GANs for distilling pretrained diffusion models is initialization. The pre-trained model serves a dual role: it is a powerful multi-step generator and an accurate estimator of the real data distribution's score. This allows it to effectively initialize all three components in the VSD framework, leading to significantly enhanced training stability.

Several methods are built upon the VSD framework, including Diff-Instruct [26], DMD [47], SID [51], and FGM [14]. The fundamental distinction between these approaches lies in the specific divergence they minimize. DMD, for instance, optimizes the reverse KL divergence between the real and generated distributions. A key advantage of this choice is its computational efficiency compared to alternatives like the Fisher divergence used in SID [51]. Specifically, during generator optimization, DMD does not require gradients to be backpropagated through the fake and teacher score estimators, whereas SID does. This does not imply the two estimators are trainable in this stage for SID, but rather reflects a difference in the computational graph. This property makes DMD more amenable to engineering implementation and scalable to large base models.

Similar to our work, TDM [27] also aimed to extend DMD to few-step distillation. However, our approach differs from TDM in three key aspects: (a) The lack of proper theoretical grounding in TDM renders its fake flow training formulation suboptimal, undermining the foundations of DMD. (b) Our framework inherently produces MoE models for few-step generation. (c) While TDM uses disjoint SNR intervals, our method employs reverse nested intervals, where each interval is a subset of the subsequent one.

B. Detailed Derivation

We show the detailed derivation of Eq. 5 as follows:

$$
\begin{aligned}
&J_{flowmatch} \\
=&E_{x_0\sim p(x_0),t\sim\mathcal{T},\boldsymbol{\epsilon}\sim\mathcal{N},x_t=\alpha_t x_0+\sigma_t\boldsymbol{\epsilon}} \\
&[\|\boldsymbol{\psi}(x_t,t)-(\boldsymbol{\epsilon}-x_0)\|^2] \qquad (14)\\
=&E_{x_0\sim p(x_0),t\sim\mathcal{T},\boldsymbol{\epsilon}\sim\mathcal{N},x_t=\alpha_t x_0+\sigma_t\boldsymbol{\epsilon}} \\
&[\|\boldsymbol{\psi}(x_t,t)-(\boldsymbol{\epsilon}-\frac{x_t-\sigma_t\boldsymbol{\epsilon}}{\alpha_t})\|^2] \qquad (15)\\
=&E_{x_0\sim p(x_0),t\sim\mathcal{T},\boldsymbol{\epsilon}\sim\mathcal{N},x_t=\alpha_t x_0+\sigma_t\boldsymbol{\epsilon}} \\
&[\|\boldsymbol{\psi}(x_t,t)+\frac{1}{\alpha_t}x_t-(1+\frac{\sigma_t}{\alpha_t})\boldsymbol{\epsilon}\|^2] \qquad (16)\\
=&E_{x_0\sim p(x_0),t\sim\mathcal{T},x_t\sim p(x_t|x_0)} \\
&[\|\boldsymbol{\psi}(x_t,t)+\frac{1}{\alpha_t}x_t+(\sigma_t+\frac{\sigma_t^2}{\alpha_t})\nabla_{x_t}\log(p(x_t|x_0))\|^2] \qquad (17)\\
=&E_{t\sim\mathcal{T},x_t\sim p(x_t)} \\
&[\|\boldsymbol{\psi}(x_t,t)+\frac{1}{\alpha_t}x_t+(\sigma_t+\frac{\sigma_t^2}{\alpha_t})\nabla_{x_t}\log(p(x_t))\|^2] \qquad (18)
\end{aligned}
$$

In the derivation, we use the score of $p(x_t|x_0)$, i.e., $\nabla_{x_t}\log(p(x_t|x_0)) = -\frac{1}{\sigma_t}\boldsymbol{\epsilon}$, and the equivalence between DSM and ESM [40].

We show the detailed derivation of Eq. 11 as follows:

$$
\begin{aligned}
&J_{subinterval-flowmatch}\\
=&E_{t\sim\mathcal{T}(t;s,1),x_t\sim p(x_t)}\\
&[\|\boldsymbol{\psi}(x_t,t)+\frac{1}{\alpha_t}x_t+(\sigma_t+\frac{\sigma_t^2}{\alpha_t})\nabla_{x_t}\log(p(x_t))\|^2] \quad (19)\\
=&E_{x_s\sim p(x_s),t\sim\mathcal{T}(t;s,1),x_t\sim p(x_t|x_s)}\\
&[\|\boldsymbol{\psi}(x_t,t)+\frac{1}{\alpha_t}x_t+(\sigma_t+\frac{\sigma_t^2}{\alpha_t})\nabla_{x_t}\log(p(x_t|x_s))\|^2] \quad (20)\\
=&E_{x_s\sim p(x_s),t\sim\mathcal{T}(t;s,1),\boldsymbol{\epsilon}\sim\mathcal{N},x_t=\alpha_{t|s}x_s+\sigma_{t|s}\boldsymbol{\epsilon}}\\
&[\|\boldsymbol{\psi}(x_t,t)+\frac{1}{\alpha_t}x_t-\frac{\sigma_t+\frac{\sigma_t^2}{\alpha_t}}{\sigma_{t|s}}\boldsymbol{\epsilon}\|^2] \quad (21)\\
=&E_{x_s\sim p(x_s),t\sim\mathcal{T}(t;s,1),\boldsymbol{\epsilon}\sim\mathcal{N},x_t=\alpha_{t|s}x_s+\sigma_{t|s}\boldsymbol{\epsilon}}\\
&[\|\boldsymbol{\psi}(x_t,t)+\frac{\alpha_{t|s}x_s+\sigma_{t|s}\boldsymbol{\epsilon}}{\alpha_t}-\frac{\sigma_t+\frac{\sigma_t^2}{\alpha_t}}{\sigma_{t|s}}\boldsymbol{\epsilon}\|^2] \quad (22)\\
=&E_{x_s\sim p(x_s),t\sim\mathcal{T}(t;s,1),\boldsymbol{\epsilon}\sim\mathcal{N},x_t=\alpha_{t|s}x_s+\sigma_{t|s}\boldsymbol{\epsilon}}\\
&[\|\boldsymbol{\psi}(x_t,t)-(\frac{\alpha_s^2\sigma_t+\alpha_t\sigma_s^2}{\alpha_s^2\sigma_{t|s}}\boldsymbol{\epsilon}-\frac{1}{\alpha_s}x_s)\|^2] \quad (23)
\end{aligned}
$$

The relationship between sample prediction (x-prediction) and score matching is derived as follows:

$$
\begin{aligned}
&J_{sample} \quad (24)\\
=&E_{x_0\sim p(x_0),t\sim\mathcal{T},\boldsymbol{\epsilon}\sim\mathcal{N},x_t=\alpha_t x_0+\sigma_t\boldsymbol{\epsilon}}[\|\boldsymbol{\mu}(x_t,t)-x_0\|^2] \quad (25)\\
=&E_{x_0\sim p(x_0),t\sim\mathcal{T},\boldsymbol{\epsilon}\sim\mathcal{N},x_t=\alpha_t x_0+\sigma_t\boldsymbol{\epsilon}}[\|\boldsymbol{\mu}(x_t,t)-\frac{x_t-\sigma_t\boldsymbol{\epsilon}}{\alpha_t}\|^2] \quad (26)\\
=&E_{x_0\sim p(x_0),t\sim\mathcal{T},\boldsymbol{\epsilon}\sim\mathcal{N},x_t=\alpha_t x_0+\sigma_t\boldsymbol{\epsilon}}\\
&[\|\boldsymbol{\mu}(x_t,t)-\frac{1}{\alpha_t}x_t+\frac{\sigma_t}{\alpha_t}\boldsymbol{\epsilon}\|^2] \quad (27)\\
=&E_{x_0\sim p(x_0),t\sim\mathcal{T},x_t\sim p(x_t|x_0)}\\
&[\|\boldsymbol{\mu}(x_t,t)-\frac{1}{\alpha_t}x_t-\frac{\sigma_t^2}{\alpha_t}\nabla_{x_t}\log(p(x_t|x_0))\|^2] \quad (28)\\
=&E_{t\sim\mathcal{T},x_t\sim p(x_t)}\\
&[\|\boldsymbol{\mu}(x_t,t)-\frac{1}{\alpha_t}x_t-\frac{\sigma_t^2}{\alpha_t}\nabla_{x_t}\log(p(x_t))\|^2] \quad (29)
\end{aligned}
$$

The training objective for x-prediction diffusion models within a subinterval is as follows:

$$
\begin{aligned}
&J_{subinterval-sample}\\
=&E_{t\sim\mathcal{T},x_t\sim p(x_t)} \quad (30)\\
&[\|\boldsymbol{\mu}(x_t,t)-\frac{1}{\alpha_t}x_t-\frac{\sigma_t^2}{\alpha_t}\nabla_{x_t}\log(p(x_t))\|^2] \quad (31)\\
=&E_{x_s\sim p(x_s),t\sim\mathcal{T}(t;s,1),x_t\sim p(x_t|x_s)}\\
&[\|\boldsymbol{\mu}(x_t,t)-\frac{1}{\alpha_t}x_t-\frac{\sigma_t^2}{\alpha_t}\nabla_{x_t}\log(p(x_t|x_s))\|^2] \quad (32)\\
=&E_{x_s\sim p(x_s),t\sim\mathcal{T}(t;s,1),\boldsymbol{\epsilon}\sim\mathcal{N},x_t=\alpha_{t|s}x_s+\sigma_{t|s}\boldsymbol{\epsilon}}\\
&[\|\boldsymbol{\mu}(x_t,t)-\frac{1}{\alpha_t}x_t+\frac{\sigma_t^2}{\alpha_t\sigma_{t|s}}\epsilon\|^2] \quad (33)\\
=&E_{x_s\sim p(x_s),t\sim\mathcal{T}(t;s,1),\boldsymbol{\epsilon}\sim\mathcal{N},x_t=\alpha_{t|s}x_s+\sigma_{t|s}\boldsymbol{\epsilon}}\\
&[\|\boldsymbol{\mu}(x_t,t)-\frac{\alpha_{t|s}x_s+\sigma_{t|s}\boldsymbol{\epsilon}}{\alpha_t}+\frac{\sigma_t^2}{\alpha_t\sigma_{t|s}}\boldsymbol{\epsilon}\|^2] \quad (34)\\
=&E_{x_s\sim p(x_s),t\sim\mathcal{T}(t;s,1),\boldsymbol{\epsilon}\sim\mathcal{N},x_t=\alpha_{t|s}x_s+\sigma_{t|s}\boldsymbol{\epsilon}}\\
&[\|\boldsymbol{\mu}(x_t,t)-(\frac{1}{\alpha_s}x_s-\frac{\alpha_t\sigma_s^2}{\alpha_s^2\sigma_{t|s}}\boldsymbol{\epsilon})\|^2] \quad (35)
\end{aligned}
$$

C. Experimental Details

For the Wan2.x base models, distillation on the text-to-image task is conducted at a fixed data resolution of one frame with width 1280 and height 720, *i.e.*, frame $=1$, width $=1280$, height $=720$.

For the Wan2.2-x2V-A14B model, distillation on both the text-to-video and image-to-video tasks employs a mixture of data resolutions: $(81, 720, 1280)$, $(81, 1280, 720)$, $(81, 480, 832)$, $(81, 832, 480)$, sampled with probabilities $0.1, 0.1, 0.4, 0.4$.

For the Qwen-Image model, distillation on the text-to-image task employs a uniform sampling from a set of resolutions: $(1, 1382, 1382)$, $(1, 1664, 928)$, $(1, 928, 1664)$, $(1, 1472, 1104)$, $(1, 1104, 1472)$, $(1, 1584, 1056)$, $(1, 1056, 1584)$.

A timestep shift of 5 is applied for Wan2.x base models, following the self-forcing approach [13] while a shift of 3 is used for the Qwen-Image base model, based on ComfyUI's Qwen-Image workflow. The resulting timesteps are provided in Tab. 1. Fig. 8 illustrates how intervals are determined by the steps and the timestep shift value.

For Wan2.2 base models, the first training phase exclusively employs the high-noise model. During this phase, the re-noising timestep t is restricted (by torch.clamp) to the range (0.875, 1) for the T2V task and (0.9, 1) for the I2V task, in accordance with the boundary timestep configuration of the high-noise model. In the second phase, both the high-noise and low-noise models are employed and three components are trainable: the low-noise generator, the high-noise fake model and the low-noise fake model. The choice of high-noise or low-noise teacher and fake model is

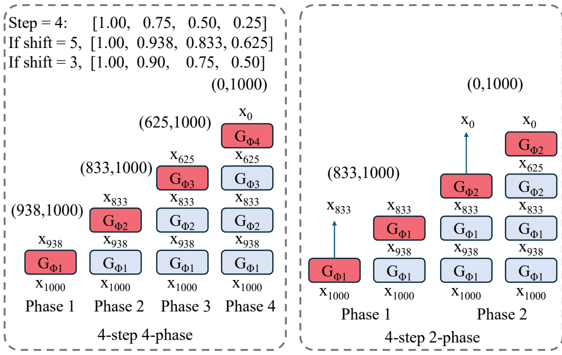

Figure 8. The timestep intervals are determined by the steps and the timestep shift value. For Wan-based models, shift = 5 is used, following the Self-Forcing approach. For Qwen-Image-based models, shift = 3 is adopted, based on ComfyUI's Qwen-Image workflow. The re-noising t range for each phase is represented as a tuple (min step, max step) and t is uniformly sampled before applying the shift. For instance, in a 4-step, 2-phase setting, t is sampled within (0.5, 1) and then apply a shift of 5 during Phase 1.

determined by the values of the re-noising timestep. This training paradigm aligns with our strategy for sampling noise injection timesteps, as detailed in Sec. D.2.

D. More Results

D.1. Additional Quantitative Evaluation on Video Generation

We present additional quantitative evaluation results in Tab. 4 and Tab. 5. Interestingly, across most evaluation dimensions, the base model using 40 inference steps (80 function evaluations) exhibits the poorest performance. For instance, in text-to-video generation, the base model achieves an aesthetic quality score of only 63.62 %, whereas both distilled variants using 4 steps (4 function evaluations) obtain higher scores. Although the Vbench [15] quantitative metrics suggest that DMD2 achieves the best overall performance while the base model performs worst, human preference ratings show the opposite trend. In the attached video, we compile all 220 generated comparison videos for the T2V task. Note that the video has been heavily compressed to reduce file size. As the video clearly demonstrates, the base model exhibits the highest overall quality in terms of aesthetics and motion dynamics, while *Phased DMD preserves the base model's performance*, substantially better than DMD2. We argue that the rankings derived from the quantitative evaluations in Tab. 4 and Tab. 5 are not entirely reliable. Nevertheless, the performance gaps between the distilled models and the base model reveal that Phased DMD produces values more closely aligned with those of the base model, indicating that Phased DMD better preserves the generative distribution of the original base model.

(a) Disjoint Intervals (b) Reverse Nested Intervals

Figure 9. The effect of noise injection intervals. Luo et al. [27] employs disjoint noise injection timestep intervals for different generation steps, where the intervals do not overlap. In contrast, we adopt reverse nested intervals, where the diffusion timestep interval in each phase terminates at 1.0. Integrating disjoint intervals into Phased DMD leads to unnatural colors and deteriorated facial structures, as illustrated on the left. Conversely, adopting reverse nested intervals yields correct results.

The results presented in Tab. 2 are based on the 4-step 2-phase configuration, as illustrated in Fig. 1d. Tab. 6 demonstrates that the 4-step 4-phase configuration (Fig. 1c) delivers better performance in both motion dynamics and visual quality, albeit at the cost of increased system complexity. This improvement can be attributed to the avoidance of SGTS and the inclusion of more trainable parameters.

D.2. Ablation on Diffusion Timestep Subintervals

Empirically, we observe that sampling noise injection timesteps using **Reverse Nested Intervals** $t \sim \mathcal{T}(t; t_k, 1)$ outperforms **Disjoint Intervals** $t \sim \mathcal{T}(t; t_k, t_{k-1})$ in terms of generation quality. Fig. 9 illustrates the results of these two methods in the Wan2.2 T2V distillation task. Specifically, sampling $t \sim \mathcal{T}(t; t_k, 1)$ yields normal color tones and accurate structures, whereas sampling $t \sim \mathcal{T}(t; t_k, t_{k-1})$ results in low-contrast tones and degraded facial structures.

At the beginning of each phase in Phased DMD, there is a substantial gap between the distribution of samples generated by the few-step generator and the distribution of real samples. The generated samples fall outside the domain of the teacher model, leading to inaccurate score estimations. This discrepancy is particularly pronounced in the

Table 4. More quantitative comparison on text-to-video generation. The base model is Wan2.2-T2V-A14B.

Method	aesthetic quality	background consistency	motion smoothness	subject consistency	temporal flickering
Base model	63.62 %	94.03 %	97.67 %	90.06 %	95.70 %
DMD2	67.02 % (+3.40 %)	95.29 % (+1.26 %)	98.57 % (+0.90 %)	92.93 % (+2.87 %)	97.08 % (+1.38 %)
Phased DMD(Ours)	65.73 % (+2.11 %)	94.40 % (+0.37 %)	97.74 % (+0.07 %)	91.15 % (+1.09 %)	95.26 % (-0.44 %)

Table 5. More quantitative comparison on image-to-video generation. The base model is Wan2.2-I2V-A14B.

Method	aesthetic quality	background consistency	motion smoothness	subject consistency	temporal flickering
Base model	62.71 %	93.75 %	97.74 %	90.39 %	95.52 %
DMD2	64.14 % (+1.43 %)	94.44 % (+0.69 %)	97.85 % (+0.11 %)	91.84 % (+1.45 %)	95.73 % (+0.21 %)
Phased DMD(Ours)	63.91 % (+1.20 %)	94.16 % (+0.41 %)	97.66 % (-0.08 %)	91.40 % (+1.01 %)	95.17 % (-0.35 %)

Table 6. Quantitative comparison of video generation performance. "OF" refers to optical flow. "DD" refers to dynamic degree. Completely removing SGTS from Phased DMD leads to improved performance.

Method	OF ↑	DD ↑	FID ↓	FVD ↓
Base model	**10.26**	79.55 %	**0.0**	**0.0**
DMD2	3.23	65.45 %	55.70	763.1
Ours (4-step 2-phase)	9.30	<u>82.27 %</u>	47.24	700.9
Ours (4-step 4-phase)	<u>9.43</u>	**83.18 %**	<u>45.40</u>	<u>578.2</u>

high-SNR (low-noise level) range, where samples are less corrupted by noise. In contrast, in the low-SNR (high-noise level) range, the diffused generated distribution overlaps more significantly with the diffused real distribution, enabling the teacher model to provide more accurate score estimations. Consequently, noise injection at high-noise levels plays a crucial role in DMD training.

To validate this analysis, we perform ablation studies on vanilla DMD for the Wan2.1 T2I task. Specifically, the diffusion timestep t is fixed at 0.357 for one experiment and at 0.882 for another. Wang et al. [43] has proven that $D_{KL}(p_{fake}(x_t)\|p_{real}(x_t)) = 0 \Leftrightarrow D_{KL}(p_{fake}(x_0)\|p_{real}(x_0)) = 0$ for any $0 < t < 1$. Thus, both experiments are theoretically valid. However, the experiment with a diffusion timestep $t = 0.357$ fails to converge, as illustrated in Fig. 10, while the experiment with $t = 0.882$ demonstrates correct results. This controlled experiment highlights that incorporating high-noise levels is essential for effective DMD training. This observation, to some extent, explains our rationale for adopting **Reverse Nested Intervals**, wherein the training interval at each stage includes the high-noise range.

(a) t = 0.357 (b) t = 0.882

Figure 10. The effect of noise injection timestep in DMD training. In DMD training, noise is injected into the generated samples at a low noise level (left) and a high noise level (right). The training fails to converge correctly when noise is injected exclusively at a low noise level.

D.3. Additional Discussion on MoE

Mixture-of-Experts (MoE) architectures are widely employed in large language models [2, 46], where they are typically adapted within the feed-forward network (FFN) layers. In diffusion models, however, MoE is implemented differently. Here, each expert typically features a dense architecture and is assigned to a specific denoising range, allowing it to be optimized for a distinct subset of the generative process. This functional division, which aligns with the

(a) low-SNR only (b) low-SNR + high-SNR

Figure 11. Visualization of the functional roles of the low-SNR (high-noise) and high-SNR (low-noise) experts. (a) Video sequences generated by the distilled high-noise expert of Wan2.2-T2V-A14B, evaluated over only the first two denoising steps. (b) Video sequences generated by the combined pipeline of the distilled high-noise and low-noise experts, evaluated over all four denoising steps. A comparison of (a) and (b) demonstrates that the low-SNR expert is responsible for modeling global structure and dynamics, whereas the high-SNR expert refines local details.

different requirements across the denoising trajectory, is illustrated in Fig. 11: the low-SNR (high-noise) stage is critical for modeling global structures and dynamics, whereas the high-SNR (low-noise) stage focuses on refining fine-grained details. During inference, these experts are applied sequentially as the SNR increases throughout the sampling process.

The scaling of diffusion models has recently increased interest in MoE architectures. By dedicating experts to different SNR levels, MoE enhances model capacity and generative quality without a proportional increase in inference cost. This performance gain is particularly pronounced in video generation [41], where a dedicated high-noise expert excels at capturing coherent temporal dynamics.

Our training framework, Phased DMD , is naturally compatible with such MoE-based models. For a base model with N experts, the optimal practice is to employ an N-phase training scheme. In the k-th phase, the setup comprises one trainable generator and k trainable fake models. Empirical observations indicate that 4-step sampling represents a performance-efficient balance. Consequently, we posit that a base model architecture with four denoising experts is an effective choice. Given the inherent compatibility between diffusion models and MoE architectures, as well as the demonstrated benefits of specialized experts, we argue that MoE will become an increasingly prevalent choice for image and video generation. Correspondingly, Phased DMD will play a more significant role in the distillation of diffusion models, as it is inherently compatible with MoE-based foundations.