

---

# DECOUPLED DMD: CFG AUGMENTATION AS THE SPEAR, DISTRIBUTION MATCHING AS THE SHIELD

Dongyang Liu<sup>1,2</sup> Peng Gao<sup>1,✉</sup> David Liu<sup>1,2</sup> Ruoyi Du<sup>1</sup> Zhen Li<sup>1</sup> Qilong Wu<sup>1</sup>  
Xin Jin<sup>1</sup> Sihan Cao<sup>1</sup> Shifeng Zhang<sup>1</sup> Hongsheng Li<sup>2,✉</sup> Steven HOI<sup>1</sup>

<sup>1</sup>Tongyi Lab, Alibaba Group    <sup>2</sup>The Chinese University of Hong Kong  
✉jingpeng\_gp@alibaba-inc.com    ✉hsli@ee.cuhk.edu.hk

⌚ <https://github.com/Tongyi-MAI/Z-Image>

## ABSTRACT

Diffusion model distillation has emerged as a powerful technique for creating efficient few-step and single-step generators. Among these, Distribution Matching Distillation (DMD) and its variants stand out for their impressive performance, which is widely attributed to their core mechanism of matching the student’s output distribution to that of a pre-trained teacher model. *In this work, we challenge this conventional understanding.* Through a rigorous decomposition of the DMD training objective, we reveal that in complex tasks like text-to-image generation, where CFG is typically required for desirable few-step performance, the primary driver of few-step distillation is not distribution matching, but a previously overlooked component we identify as *CFG Augmentation (CA)*. We demonstrate that this term acts as the core “engine” of distillation, while the **Distribution Matching (DM)** term functions as a “regularizer” that ensures training stability and mitigates artifacts. We further validate this decoupling by demonstrating that while the DM term is a highly effective regularizer, it is not unique; simpler non-parametric constraints or GAN-based objectives can serve the same stabilizing function, albeit with different trade-offs. This decoupling of labor motivates a more principled analysis of the properties of both terms, leading to a more systematic and in-depth understanding. This new understanding further enables us to propose principled modifications to the distillation process, such as decoupling the noise schedules for the engine and the regularizer, leading to further performance gains. Notably, our method has been adopted by the **Z-Image** project to develop a top-tier 8-step image generation model, empirically validating the generalization and robustness of our findings.

## 1 INTRODUCTION

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020) have rapidly risen to prominence in generative modeling, achieving state-of-the-art performance and producing images of unprecedented quality and diversity. This success has sparked widespread interest across both academia and industry. However, the power of these models comes at a significant cost: their iterative sampling process, often requiring dozens to hundreds of neural network evaluations, is computationally expensive and slow, hindering their use in real-time applications.

To address this limitation, a flurry of research has focused on converting the original diffusion model into a few-step generator. Typical technical routes include *direct distillation* (Liu et al., 2022), where the student is expected to replicate the trajectory of the teacher; *consistency distillation* (Song et al., 2023), which enforces self-consistency along the sampling trajectory; and *adversarial distillation* (Sauer et al., 2024b), which leverages an adversarial objective to match the student’s output distribution to target, showing impressive results in high-resolution synthesis.

Among these diverse approaches, score-based distillation, notably represented by Diff-Instruct (Luo et al., 2023b), Distribution Matching Distillation (DMD) (Yin et al., 2024b), and other variants (Yin et al., 2024a; Chadebec et al., 2025), has been recognized as exceptionally promising. Its advantages are twofold: not only does it achieve state-of-the-art performance, but it is also underpinned by

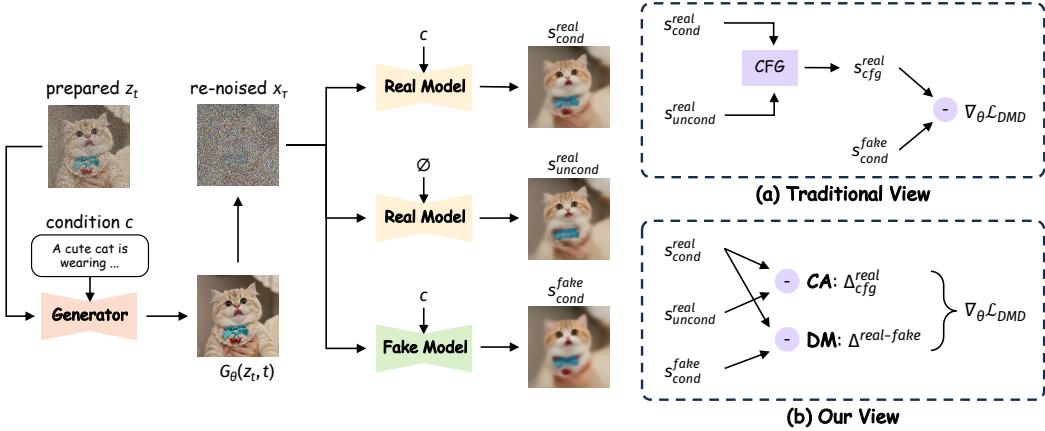


Figure 1: Two perspectives on the DMD algorithm. (a) The conventional view, which treats the use of CFG as a heuristic relaxation of the theoretical framework, with the algorithm’s success solely attributed to this (relaxed) distribution matching mechanism. (b) Our proposed decoupled view, where the objective is a combination of two distinct mechanisms: a CFG Augmentation (CA) engine that drives the few-step conversion, and a Distribution Matching (DM) regularizer—which strictly adheres to the theoretical derivation (Eq. 1)—that ensures training stability.

an elegant theoretical framework. Specifically, the method is framed as minimizing an Integral Kullback-Leibler (IKL) divergence (Luo et al., 2023b) between the student’s output distribution ( $p_{\text{fake}}$ ) and the teacher’s target distribution ( $p_{\text{real}}$ ):

$$\mathcal{L}_{\text{IKL}}(p_{\text{real}}, p_{\text{fake}}) = \int_0^1 \mathbb{KL}(p_{\text{real},\tau} \parallel p_{\text{fake},\tau}) d\tau. \quad (1)$$

In practice, the gradient of this objective is estimated using a pair of “real” (teacher) and “fake” (student-tracking) models, making the training process feasible.

**However, a dark cloud has loomed over the interpretation of this elegant method: the use of Classifier-Free Guidance (CFG) (Ho & Salimans, 2022) on the real model.** According to the theoretical derivation, the ideal estimator for the target real score should be the prediction from the real model itself, with no involvement of CFG. However, empirical evidence overwhelmingly shows that on complex tasks like text-to-image, DMD-like methods yield good results only with a high CFG scale. Even if we were to boldly assume—an assumption we find to be insufficiently grounded—that CFG somehow produces a higher-quality substitute for the original score, the asymmetric application of CFG, in which only the real model but not the fake model is equipped with CFG, still creates a stark inconsistency between theory and practice. Overall, the usage of CFG breaks the integrity of the original, rigorous theoretical derivation of matching two distributions.

This strongly suggests that **the current understanding of DMD’s success is likely incomplete or inaccurate**. In this paper, we aim to redefine the understanding of how DMD and similar algorithms work. Through a rigorous decomposition of the practical DMD training objective that utilizes real score CFG, we reveal that its effectiveness is not driven by a single mechanism, but by a clear division of labor between two distinct components:

**1. CFG Augmentation (CA):** A previously overlooked term that directly applies the CFG signal to the student’s output. We demonstrate that this component acts as the core **engine** of distillation, responsible for converting a multi-step model into a high-quality few-step generator.

**2. Distribution Matching (DM):** A mechanism that perfectly aligns with the theoretical derivation (Eq. 1). While existing works have proved its independent distillation capability in simple tasks like low-resolution CIFAR (Luo et al., 2023b), we show that for complex tasks, its primary function converts more to a powerful **regularizer** that ensures training stability and mitigates artifacts.

This decoupled framework challenges the prevailing narrative and provides a more accurate explanation for the success of DMD-like methods. We substantiate our claims through a series of carefully designed experiments, including independent investigations of the effect of each component,

---

and demonstrating that the DM regularizer, while highly effective, can be conceptually replaced by simpler statistical constraints or more complex GANs. This explicit decoupling also enables a more principled and in-depth analysis of the properties and inner workings of each component. Finally, armed with this deeper understanding, we propose principled improvement by proposing decoupled renoising schedules for CA and DM, respectively, leading to further performance gains, and demonstrating the practical value of our new perspective.

## 2 RELATED WORK

**Few-Step Diffusion Distillation** aims to reduce the inference cost of diffusion models. *Trajectory-matching* approaches train a student model to replicate the teacher’s sampling path in fewer steps (Liu et al., 2023; Zhu et al., 2024; Kim et al., 2024; Frans et al., 2024; Salimans & Ho, 2022; Meng et al., 2023), with consistency distillation as a renowned branch (Song et al., 2023; Kim et al., 2023; Lu & Song, 2024; Wang et al., 2024; Ren et al., 2024). Another prominent direction is *GAN-based distillation* (Sauer et al., 2024b;a; Lin et al., 2024), which leverages an adversarial objective to match the student’s output distribution with the teacher’s or with real data.

**Score-based Distillation** was initially proposed for 3D generation (Poole et al., 2022; Wang et al., 2023). Diff-Instruct (Luo et al., 2023b) pioneered its application in few-step diffusion distillation, and DMD (Yin et al., 2024b) was among the first to successfully apply this principle to large-scale text-to-image models. Following works have explored different distribution metrics (Zhou et al., 2024b;a) or combining this principle with other distillation paradigms (Yin et al., 2024a; Chadebec et al., 2025; Luo et al., 2024). Notably, the adoption of CFG in real score is a common practice among these works, but this choice is rarely officially discussed. An exception is (Luo, 2024), which models the CFG term as an extra reward function after distillation. We are the first to decouple the role of this CFG term during distillation and to reveal its dominance in multi-to-few-step conversion.

## 3 REVISITING AND DECOMPOSING DMD

The goal of Distribution Matching Distillation (DMD) is to train a student generator, denoted as  $G_\theta$ , to emulate the output distribution of a pre-trained, frozen teacher diffusion model in a few-step or even single-step inference process. The training is guided by minimizing a loss function, Eq. 1, whose gradient with respect to the generator’s parameters  $\theta$  can be estimated by:

$$\nabla_\theta \mathcal{L}_{\text{DMD-theory}} = \mathbb{E}_{z_t, \tau, \mathbf{x}_\tau} \left[ - \left( s_{\text{cond}}^{\text{real}}(\mathbf{x}_\tau) - s_{\text{cond}}^{\text{fake}}(\mathbf{x}_\tau) \right) \frac{\partial G_\theta(z_t)}{\partial \theta} \right]. \quad (2)$$

In this paper, we follow the flow matching notations (Lipman et al., 2022) and define  $t = 0$  with pure noise and  $t = 1$  with clean data.  $z_t$  denotes the prepared generator input at noise level  $t$ . For single-step generation,  $t$  is 0 and  $z_t$  is random noise; for few-step generation,  $z_t$  can be obtained by going through the previous steps, a technique called “backward simulation” (Yin et al., 2024a). The generator  $G_\theta$  takes  $z_t$  and makes the image prediction  $G_\theta(z_t)$ , which is then *renoised* to  $x_\tau$  with a sampled noise level  $\tau$ . After renoising,  $x_\tau$  would be fed to two diffusion models for score estimates:  $s_{\text{cond}}^{\text{real}}$ , the “real score” estimated by the original multi-step teacher model; and  $s_{\text{cond}}^{\text{fake}}$ , the “fake” score estimate from an auxiliary “fake” model that is trained concurrently on the generator’s outputs. The subscript “cond” indicates the score is conditioned on a text input. Pseudo-code provided in Sec. B

However, Eq. 2 usually leads to poor performance in practice, and a subtle modification is involved:

$$\nabla_\theta \mathcal{L}_{\text{DMD}} = \mathbb{E}_{z_t, \tau, \mathbf{x}_\tau} \left[ - \left( s_{\text{cfg}}^{\text{real}}(\mathbf{x}_\tau) - s_{\text{cond}}^{\text{fake}}(\mathbf{x}_\tau) \right) \frac{\partial G_\theta(z_t)}{\partial \theta} \right]. \quad (3)$$

The only difference between Eq. 2 and Eq. 3 is that the real score  $s_{\text{cond}}^{\text{real}}$  is replaced with  $s_{\text{cfg}}^{\text{real}}$ , where

$$s_{\text{cfg}}^{\text{real}}(\mathbf{x}_\tau) = s_{\text{uncond}}^{\text{real}}(\mathbf{x}_\tau) + \alpha \left( s_{\text{cond}}^{\text{real}}(\mathbf{x}_\tau) - s_{\text{uncond}}^{\text{real}}(\mathbf{x}_\tau) \right). \quad (4)$$

$s_{\text{cond}}^{\text{real}}$  and  $s_{\text{uncond}}^{\text{real}}$  are the conditional and unconditional scores from the real model, respectively, and  $\alpha$  is the CFG guidance scale (typically  $\alpha > 1$ ). Despite the introduction of discrepancy between theory and practice, this modification empirically yields substantially better results. Interestingly, this substitution has been largely overlooked in prior literature, often dismissed as a mere implementation detail rather than a fundamental deviation from the original theory. **However, we will show**

---

**that the seemingly minor CFG detail actually signifies a fundamentally different mechanism independent to distribution matching.**

For clarity, in the rest of the paper, we use the acronym “DMD” to specially refer to in-practice-used and CFG-involved algorithm as defined in Eq. 3. In contrast, we use the term “Distribution Matching” or its abbreviation “DM” to refer exclusively to the theoretical principle of matching two distributions, which strictly adheres to the formulation in Eq. 1 and Eq. 2. We will show that the success of DMD is from the cooperation of two different mechanisms, with Distribution Matching playing a crucial, yet secondary, role as a regularizer rather than the primary distillation engine.

### 3.1 DECOMPOSING THE DMD GRADIENT

To scrutinize the underlying mechanisms of the DMD algorithm, we begin by substituting the definition of Classifier-Free Guidance (Eq. 4) into the DMD gradient formula (Eq. 3):

$$\nabla_{\theta} \mathcal{L}_{\text{DMD}} = \mathbb{E} \left[ - \left[ s_{\text{uncond}}^{\text{real}}(\mathbf{x}_\tau) + \alpha \left( s_{\text{cond}}^{\text{real}}(\mathbf{x}_\tau) - s_{\text{uncond}}^{\text{real}}(\mathbf{x}_\tau) \right) - s_{\text{cond}}^{\text{fake}}(\mathbf{x}_\tau) \right] \frac{\partial G_{\theta}(z_t)}{\partial \theta} \right]. \quad (5)$$

With simple rearrangement, we can decompose Eq. 5 into two distinct components:

$$\nabla_{\theta} \mathcal{L}_{\text{DMD}} = \mathbb{E} \left[ - \left( \underbrace{\left( s_{\text{cond}}^{\text{real}}(\mathbf{x}_\tau) - s_{\text{cond}}^{\text{fake}}(\mathbf{x}_\tau) \right)}_{\Delta^{\text{real-fake}} \text{ (Distribution Matching)}} + (\alpha - 1) \underbrace{\left( s_{\text{cond}}^{\text{real}}(\mathbf{x}_\tau) - s_{\text{uncond}}^{\text{real}}(\mathbf{x}_\tau) \right)}_{\Delta_{\text{cfg}}^{\text{real}} \text{ (CFG Augmentation)}} \right) \frac{\partial G_{\theta}(z_t)}{\partial \theta} \right]. \quad (6)$$

This decomposition reframes the DMD objective as a sum of two terms:

1. **Distribution Matching (DM,  $\Delta^{\text{real-fake}}$ )**: The first term,  $s_{\text{cond}}^{\text{real}} - s_{\text{cond}}^{\text{fake}}$ , strictly aligns with theoretical deduction of matching two distributions (Eq. 1 and 2).
2. **CFG Augmentation (CA,  $\Delta_{\text{cfg}}^{\text{real}}$ )**: The second term,  $(\alpha - 1)(s_{\text{cond}}^{\text{real}} - s_{\text{uncond}}^{\text{real}})$ , directly applies a scaled CFG signal as a gradient to the student’s output. This component was typically overlooked.

This separation motivates an ablation study to isolate the true contribution of each component. We investigate three training configurations: (1) the full DMD objective ( $\Delta^{\text{real-fake}} + \Delta_{\text{cfg}}^{\text{real}}$ ), (2) CFG Augmentation only ( $\Delta_{\text{cfg}}^{\text{real}}$ ), and (3) Distribution Matching only ( $\Delta^{\text{real-fake}}$ ).

#### 3.1.1 ABLATION STUDY: ENGINE VS. REGULARIZER

As illustrated in Fig. 2, our experiments reveal a clear division of labor between the two components. Training with CA alone is remarkably effective at converting the multi-step model into a few-step generator. Besides, the generated results also demonstrate high similarity in content to the full DMD objective, indicating the dominant role of the CA term in DMD loss. In contrast, even though it is improper to conclude that the DM term is totally incapable of doing the multi-step to few-step conversion (since in the 4-step experiment it indeed makes relatively reasonable images), a significant performance gap exists towards the CA setting, as indicated by both image visualizations and numerical indicators (Image Reward (Xu et al., 2023) and HPS v2.1 (Wu et al., 2023)).

However, we also observe that training with CA alone is unsustainable. While initially effective, the generated images progressively suffer from artifacts such as over-saturation and high-frequency noise, eventually leading to training collapse. The introduction of the Distribution Matching term eliminates these issues, enabling stable training over extended periods and yielding higher-quality final results. These empirical findings lead to two fundamental conclusions:

1. **CFG Augmentation is the engine for few-step conversion.** The ability of the distilled generator to produce high-quality samples in a few steps is almost entirely attributable to the  $\Delta_{\text{cfg}}^{\text{real}}$  term.
2. **Distribution Matching is a regularizer for training stability.** The  $\Delta^{\text{real-fake}}$  term, while not the primary driver of distillation, plays a crucial role as a regularizer that prevents the training process from diverging and ensures the quality of the final output.

This insight fundamentally challenges the prevailing understanding of DMD-like methods: the conversion to a few-step generator is not primarily an act of matching distributions but rather **a direct consequence of “baking” the CFG pattern into the student generator’s predictions** (we elaborate on this point in Sec. A), which is irrelevant to the fake model.



Figure 2: Ablation study on the roles of CFG Augmentation (CA) and Distribution Matching (DM). Numerical indicators are evaluated on 1k sampled prompts from COCO-10k (Lin et al., 2014).

### 3.2 DISTRIBUTION MATCHING: A GOOD, BUT NOT THE ONLY, REGULARIZER

To further validate the aforementioned division between engine and regularizer, we investigate whether alternative regularization schemes can effectively stabilize the CFG Augmentation (CA) engine. This section demonstrates that even a simple non-parametric statistical constraint can prevent training collapse, thereby substantiating the role of Distribution Matching (DM) as a regularizer. Concurrently, we highlight that DM is an exceptionally well-suited regularizer for this task, exhibiting clear advantages over both simpler non-parametric methods and more complex GAN-based approaches.

**Non-Parametric Mean-Variance Regularization.** As shown in Fig. 3, training with the CA engine leads to a monotonic increase in the variance of generated images, finally reaching unreasonably large values. This inspires us to design the simplest regularization term of constraining the mean and variance of the generator’s output. Specifically, we apply a Kullback-Leibler (KL) divergence loss that aligns the per-image mean  $\mu$  and variance  $\sigma^2$  of the student’s output  $G_\theta(z_t)$  with target

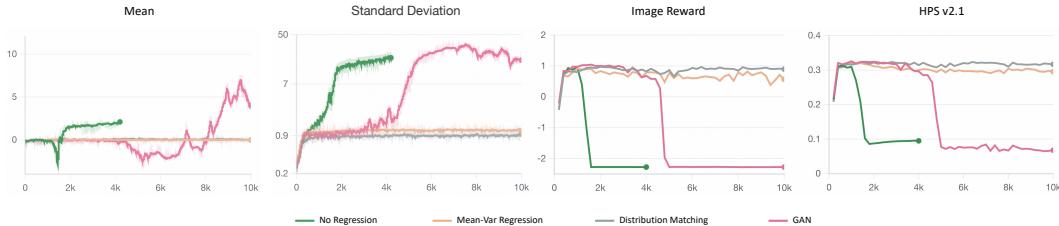


Figure 3: CFG Augmentation with different regularizers. Image Reward and HPS v2.1 evaluated on 1k sampled prompts from COCO-10k. Setting: 4-step SDXL. See Fig. 6 for visualized samples.

statistics  $(\mu_{\text{target}}, \sigma_{\text{target}}^2)$ . For SDXL experiments, we use  $\mu_{\text{target}} = 0.075$  and  $\sigma_{\text{target}}^2 = 0.81$ , which are the averaged statistics of sampled real data. For a batch of  $B$  generated images, the loss is defined as:

$$\mathcal{L}_{\text{KL}} = \frac{1}{B} \sum_{i=1}^B \frac{1}{2} \left( \frac{\sigma_i^2 + (\mu_i - \mu_{\text{target}})^2}{\sigma_{\text{target}}^2} - 1 - \log \frac{\sigma_i^2}{\sigma_{\text{target}}^2} \right), \quad (7)$$

where  $(\mu_i, \sigma_i^2)$  are the mean and variance of the  $i$ -th generated image.

As shown in Fig. 3 and Fig. 6, this simplest non-parametric regularization proves remarkably effective at stabilizing the training process, allowing the CA engine to operate durably, keeping the quality indicators at a relatively high level. This result strongly reinforces the hypothesis that the primary role of the DM term is a regularizer. However, the final image quality, while stable, falls noticeably short of that achieved with DM. This suggests that the artifacts induced by the CA engine are more complex than what can be captured by mean&variance alone.

**GAN-based Regularization** A more powerful candidate for regularization is a GAN discriminator, a technique already employed in several diffusion distillation works (Yin et al., 2024a; Lin et al., 2025). Following existing methods (Sauer et al., 2024b), we use a discriminator initialized from the weights of the pre-trained teacher model. Experiments (Fig. 3) confirm that a GAN can indeed function as a regularizer, effectively controlling image variance and eliminating certain artifacts. However, GAN requires image data during the distillation process. Besides, the approach introduces significant challenges in training stability, as the training still collapsed after 4k iterations.

Our investigation suggests that while distribution matching is not the only possible regularizer, it represents a sweet spot—offering a more powerful corrective signal than simple statistical constraints, while being substantially more stable and less complex than GANs. The comparison suggests a trade-off between stability and potential performance. The increasing complexity from statistical constraints to GANs correlates with decreasing training robustness, which echoes the claim in Diff-Instruct (Luo et al., 2023b) that score-matching can be viewed as a more stable alternative to GANs, especially when the distributions have disjoint supports. Conversely, greater complexity may offer a higher performance ceiling. This aligns with practices in VAE training (Rombach et al., 2021; 2022) or advanced few-step distillation (Lin et al., 2025), where models are often first trained with a stable objective before being fine-tuned with a GAN loss to achieve peak performance.

## 4 MECHANISTIC ANALYSIS OF CA AND DM

The decomposition of the DMD objective into a CA engine and a DM regularizer enables an in-depth exploration of their respective inner workings. This section aims to answer two fundamental questions: first, how exactly does the CA engine drive the multi-step to few-step conversion? And second, by what mechanism does the DM regularizer ensure the stability of this process?

### 4.1 DISSECTING THE CA ENGINE: THE ROLE OF THE RE-NOISING SCHEDULE

To understand the working mechanism of the CA engine, we investigate a central question: **How does the choice of re-noising timestep  $\tau$  influence the effect of CA?** Since  $\tau$  determines the noise level at which the CFG signal is computed, it serves as a powerful probe into the engine’s behavior.

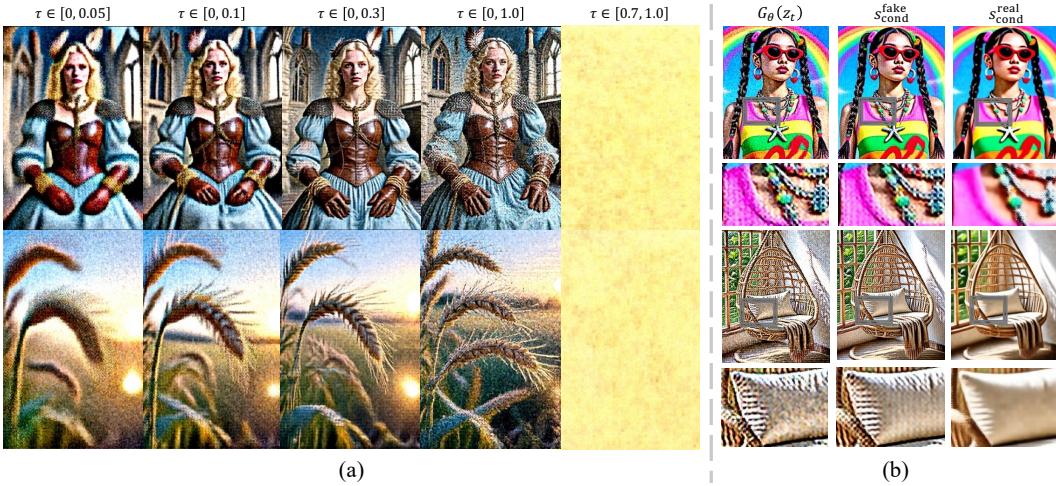


Figure 4: (a) Visualization on the effect of re-noising timestep  $\tau$  in CFG Augmentation (CA). The generator is trained with CA alone. In our notation,  $\tau = 0$  corresponds to pure noise and  $\tau = 1$  to clean data. (b) Illustration of the DM corrective mechanism. The generator is trained with CA alone, while the fake model keeps training on the generator’s output as in DMD.

To isolate this effect, we design an experiment where a single-step generator is trained using only the CA term ( $\Delta_{\text{cfg}}^{\text{real}}$ ). We then systematically vary the sampling range of the re-noising timestep  $\tau$ , starting from the noisiest end of the spectrum and gradually expanding towards cleaner timesteps.

Results in Fig. 4 (a) reveal a clear and consistent pattern. When  $\tau$  is restricted to a noisy range (e.g.,  $[0.0, 0.05]$ ), the CA engine primarily enhances low-frequency information, such as broad color blocks and overall composition. As the range of  $\tau$  expands to include cleaner timesteps, the generated images progressively gain richer, higher-frequency details, like sharp edges and fine textures. This leads to a crucial conclusion: **the CA engine, when applied at a specific noise level  $\tau$ , primarily enhances the image content corresponding to that level**. This conclusion is further supported by the fact that the training collapses when  $\tau$  is restricted to clean timesteps only ( $[0.7, 1.0]$ ): high-frequency details are meaningless if low-frequency general structure has not yet been determined.

This finding has a critical implication for multi-step generation. Consider a generator at step  $t$  operating on an input  $z_t$ , which already contains resolved information for noise levels below  $t$ . Is it still necessary, or even detrimental, to apply CA with a re-noising range containing  $\tau < t$ ? Our analysis suggests this would be redundant, potentially over-enhancing already established features and leading to artifacts. We therefore hypothesize that **an optimal CA schedule should act as a focused engine, concentrating its power on the remaining, unresolved aspects of the image by constraining its re-noising schedule to  $\tau > t$** . Experimental validation is provided in Sec. 4.3.

#### 4.2 UNDERSTANDING THE DM REGULARIZER: A CORRECTIVE MECHANISM

Having established CA as the engine, we now turn to the DM regularizer and ask: **How does it counteract the artifacts introduced by CA and ensure training stability?**

To gain insight into this corrective mechanism, we design a specific diagnostic experiment. We continue to train the generator using only the CA engine, a setting we know is unstable and produces artifacts. However, we introduce an auxiliary “fake” model as a non-interfering **observer**. This fake model is trained concurrently on the generator’s outputs—just as in standard DMD—but its score estimate is **not** used to update the generator. This setup allows us to witness *when artifacts occur, how a potential DM gradient ( $s_{\text{cond}}^{\text{real}} - s_{\text{cond}}^{\text{fake}}$ ) would act to correct them*.

Fig. 4 (b) offers an informative observation. The image generated by the CA-only generator exhibits clear high-frequency checkerboard artifacts. When this artifact-laden image is re-noised and fed to the two score models, the artifact is conspicuously absent in the prediction from the frozen real model, yet it persists in the prediction from the observer fake model. This occurs because the fake model, by

Table 1: Ablation on different re-noising schedule configurations with Lumina-Image-2.0. Detailed results on the HPS benchmarks are provided in Tab. 6 and 7

Model	DPG Bench						HPS v2.1 Overall	HPS V3 Overall
	Global	Entity	Attribute	Relation	Other	Overall		
Original (50 steps)	84.50	90.89	91.20	94.42	86.34	87.20	30.20	9.62
① $\tau_{CA} = \tau_{DM} \in [0, 1]$ (DMD)	80.22	<u>90.45</u>	90.47	89.36	<u>91.36</u>	83.90	30.61	10.34
② $\tau_{CA} \in [0, 1], \tau_{DM} \in [0, 1]$	91.88	89.03	90.08	<u>89.51</u>	<u>88.85</u>	83.77	30.69	10.32
③ $\tau_{CA} > t, \tau_{DM} > t$	<b>93.47</b>	90.18	<u>90.94</u>	89.37	90.71	85.64	<u>31.71</u>	11.08
④ $\tau_{CA} > t, \tau_{DM} \in [0, 1]$	91.40	<b>91.62</b>	<b>91.18</b>	<b>91.93</b>	<b>91.98</b>	<b>85.85</b>	<b>32.29</b>	<b>11.59</b>



Figure 5: Un-cherry-picked qualitative comparison of different re-noising schedule configurations. Top row: ② Decoupled-Full,  $\tau_{CA}, \tau_{DM} \in [0, 1]$ . Middle row: ③ Coupled-Constrained,  $\tau_{CA}, \tau_{DM} > t$ . Bottom row: ④ our proposed Decoupled-Hybrid,  $\tau_{CA} > t, \tau_{DM} \in [0, 1]$ .

tracking the generator’s output distribution, has learned to replicate its characteristic failure modes. Consequently, in the DM gradient, the artifacts present in the fake prediction ( $s_{\text{cond}}^{\text{fake}}$ ) form a negative term. Applying this gradient to the generator’s output would thus encourage a change that actively cancels out these artifacts. This provides a concrete illustration of DM’s corrective mechanism.

This understanding also clarifies the role of the re-noising timestep  $\tau$  within the DM regularizer: it controls the **scope of correction**. A large  $\tau$  (cleaner image) allows the real and fake scores to diverge primarily on subtle, high-frequency details. In contrast, a small  $\tau$  (noisier image) destroys most details, forcing the scores to diverge on more fundamental, low-frequency elements like composition and color. This gives the DM regularizer an opportunity to correct global issues.

This leads to our final hypothesis regarding the optimal renoising schedule for DM in a few-step setting. Even late-stage generation steps, which primarily add high-frequency details, can still suffer from low-frequency artifacts like color oversaturation, either inherited from previous steps or induced by an imperfect CA schedule. To address these global issues, the DM regularizer requires a global perspective. Therefore, we propose that **the optimal DM schedule is different from that of CA: DM should function as a comprehensive regularizer, always spanning the full noise range ( $\tau_{DM} \in [0, 1]$ ), irrespective of the generator’s current timestep  $t$** .

#### 4.3 VALIDATING THE DECOUPLED SCHEDULE HYPOTHESIS

Our mechanistic analysis in Sec. 4.1 and Sec. 4.2 has led to a central hypothesis: for optimal few-step distillation, the CA engine and the DM regularizer require distinct, decoupled re-noising schedules. Specifically, we proposed that the CA schedule should be constrained ( $\tau_{CA} > t$ ) to act as a focused engine, while the DM schedule should remain global ( $\tau_{DM} \in [0, 1]$ ) to serve as a comprehensive regularizer. In this section, we empirically validate this proposal.

Table 2: Comparison of 4-step SDXL distillation. Our ‘Decoupled’ method strictly follows the DMD2 training setting but employs our proposed decoupled-hybrid (④) re-noising schedule. Indicators are evaluated on 10k COCO2014-val prompts.

Method	FID↓	CLIP-S ↑	ImageReward ↑	HPS V2.1 ↑	HPS V3 ↑
LCM (Luo et al., 2023a)	22.27	31.71	39.56	28.00	6.45
Turbo (Sauer et al., 2024b)	27.27	32.16	46.09	29.83	9.09
Lightning (Lin et al., 2024)	24.49	32.31	57.48	30.30	9.48
Flash (Chadebec et al., 2025)	22.96	31.84	19.04	27.71	6.49
PCM (Wang et al., 2024)	24.13	32.52	64.73	<b>30.76</b>	9.46
DMD2 (Yin et al., 2024a)	<u>18.95</u>	<u>33.14</u>	<u>71.01</u>	<u>30.64</u>	<u>9.64</u>
Decoupled (Ours)	<b>17.80</b>	<b>33.62</b>	<b>78.61</b>	<u>30.34</u>	<b>9.79</b>

To facilitate this investigation, we first generalize the DMD gradient (Eq.6) to a  $\tau$ -decoupled form. This allows us to assign independent re-noising schedules,  $\tau_{\text{CA}}$  and  $\tau_{\text{DM}}$ , to the CA and DM components, respectively. The resulting “decoupled DMD” (d-DMD) gradient is formulated as:

$$\nabla_{\theta} \mathcal{L}_{\text{d-DMD}} = \mathbb{E} \left[ - \left( (s_{\text{cond}}^{\text{real}}(\mathbf{x}_{\tau_{\text{DM}}}) - s_{\text{cond}}^{\text{fake}}(\mathbf{x}_{\tau_{\text{DM}}})) + (\alpha - 1) (s_{\text{cond}}^{\text{real}}(\mathbf{x}_{\tau_{\text{CA}}}) - s_{\text{uncond}}^{\text{real}}(\mathbf{x}_{\tau_{\text{CA}}})) \right) \frac{\partial G_{\theta}(z_t)}{\partial \theta} \right], \quad (8)$$

where d-DMD is short for decoupled DMD. This modification allows us to decouple the renoising schedule of DM and CA, allowing principled experimental analysis. With this formulation, we design an ablation study to evaluate four distinct schedule configurations for a 4-step generator:

- ① **Coupled-Shared:** The original DMD approach where  $\tau_{\text{CA}} = \tau_{\text{DM}}$ , sampled from  $[0, 1]$ .
- ② **Decoupled-Full:** Both schedules are independent but cover the full range,  $\tau_{\text{CA}}, \tau_{\text{DM}} \in [0, 1]$ .
- ③ **Decoupled-Constrained:** Both schedules are independent and constrained,  $\tau_{\text{CA}}, \tau_{\text{DM}} > t$ .
- ④ **Decoupled-Hybrid:** The engine is constrained while the regularizer is not,  $\tau_{\text{CA}} > t, \tau_{\text{DM}} \in [0, 1]$ .

The results, presented in Tab. 1 for the Lumina-Image-2.0 model (Qin et al., 2025), provide strong evidence for our hypothesis. First, we confirm that merely decoupling the schedules while keeping them global ② yields negligible impact compared to the baseline ①, demonstrating that the benefit comes from the schedule’s range, not just its independence. More importantly, both configurations with constrained schedules (③ and ④) significantly outperform the baselines across multiple benchmarks (Hu et al., 2024; Wu et al., 2023; Ma et al., 2025). Crucially, our proposed Decoupled-Hybrid setting ④ consistently achieves the best overall scores, validating our core proposal.

The qualitative results in Fig. 5 offer further visual confirmation. Compared to the global schedule ②, top row), constraining the CA engine (③, middle row) introduces richer, finer details, confirming the benefit of a focused engine. However, this configuration still suffers from color oversaturation, a low-frequency artifact that its constrained DM regularizer fails to correct. In stark contrast, our Decoupled-Hybrid setting (④, bottom row) retains these enhanced details while effectively mitigating the saturation artifacts, yielding the most visually appealing and natural-looking images. These observations are decisively corroborated by a comprehensive **user study** (Sec. C), where model ④ achieved a unanimous 100% preference rate in model-level comparisons. 15 annotators consistently justified their choice by its ability to generate richer details, a more realistic and less “greasy” appearance, and fewer structural deformities. Furthermore, in a three-way image-level ranking, model ④ was ranked first in 59.8% of cases, significantly outperforming the next-best model (③ at 33.8%).

Experiments on SDXL (Tab. 2) situate our findings within the broader landscape. For a rigorous comparison with DMD2 (Yin et al., 2024a), we adopted their exact training configuration, including the GAN loss, and only replaced their re-noising schedule with our Decoupled-Hybrid approach. The results demonstrate a clear advantage, confirming the effectiveness of our proposed schedule.

In summary, these results strongly validate that our engine-regularizer decomposition not only provides a deeper understanding but also unlocks tangible performance improvements, demonstrating the practical value of our new perspective.

---

## 5 CONCLUSION AND LIMITATIONS

In this work, we challenge the conventional understanding of DMD practice in complex tasks like text-to-image, revealing a functional decoupling with **CFG Augmentation (CA)** as the primary engine for few-step conversion and **Distribution Matching (DM)** as the regularizer. This new perspective allowed us to observe the distinct properties of each component and propose a principled improvement—a decoupled re-noising schedule.

However, a fundamental question remains unanswered: why does CA possess such a remarkable ability to convert a diffusion model into a few-step generator? We find that providing a precise answer is highly challenging, partly because the mechanism of CFG itself remains largely enigmatic. For interested readers, we share our high-level, preliminary understanding and explanation of this issue in Sec. A. Nevertheless, we acknowledge that a significant gap remains towards a rigorously accurate explanation, and we intend to explore this topic further in our future work.

---

## REFERENCES

- Clement Chadebec, Onur Tasar, Eyal Benaroche, and Benjamin Aubin. Flash diffusion: Accelerating any conditional diffusion model for few steps image generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 15686–15695, 2025.
- Sander Dieleman. Diffusion is spectral autoregression, 2024. URL <https://sander.ai/2024/09/02/spectral-autoregression.html>.
- Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024.
- Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.
- Beomsu Kim, Yu-Guan Hsieh, Michal Klein, Marco Cuturi, Jong Chul Ye, Bahjat Kawar, and James Thornton. Simple reflow: Improved techniques for fast flow models. *arXiv preprint arXiv:2410.07815*, 2024.
- Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.
- Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.
- Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion adversarial post-training for one-step video generation. *arXiv preprint arXiv:2501.08316*, 2025.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.
- Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.
- Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2023.
- Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024.
- 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. *arXiv preprint arXiv:2311.05556*, 2023a.
- Weijian Luo. Diff-instruct++: Training one-step text-to-image generator model to align with human preferences. *arXiv preprint arXiv:2410.18881*, 2024.
- Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36:76525–76546, 2023b.
- Weijian Luo, Colin Zhang, Debing Zhang, and Zhengyang Geng. David and goliath: Small one-step model beats large diffusion with score post-training. *arXiv preprint arXiv:2410.20898*, 2024.

- 
- Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. [arXiv preprint arXiv:2508.03789](#), 2025.
- Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In [Proceedings of the IEEE/CVF conference on computer vision and pattern recognition](#), pp. 14297–14306, 2023.
- Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. [arXiv preprint arXiv:2209.14988](#), 2022.
- Qi Qin, Le Zhuo, Yi Xin, Ruoyi Du, Zhen Li, Bin Fu, Yiting Lu, Jiakang Yuan, Xinyue Li, Dongyang Liu, et al. Lumina-image 2.0: A unified and efficient image generative framework. [arXiv preprint arXiv:2503.21758](#), 2025.
- Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis. [Advances in Neural Information Processing Systems](#), 37:117340–117362, 2024.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.
- Robin Rombach et al. latent-diffusion github repository. <https://github.com/CompVis/latent-diffusion>, 2022.
- Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. [arXiv preprint arXiv:2202.00512](#), 2022.
- Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In [SIGGRAPH Asia 2024 Conference Papers](#), pp. 1–11, 2024a.
- Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In [European Conference on Computer Vision](#), pp. 87–103. Springer, 2024b.
- Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In [International conference on machine learning](#), pp. 2256–2265. pmlr, 2015.
- Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. [arXiv preprint arXiv:2011.13456](#), 2020.
- Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023.
- Fu-Yun Wang, Zhaoyang Huang, Alexander Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, et al. Phased consistency models. [Advances in neural information processing systems](#), 37:83951–84009, 2024.
- Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Pro-lifedreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. [Advances in neural information processing systems](#), 36:8406–8441, 2023.
- Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. [arXiv preprint arXiv:2306.09341](#), 2023.
- Jiazhen Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. [Advances in Neural Information Processing Systems](#), 36:15903–15935, 2023.
- Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. [Advances in neural information processing systems](#), 37:47455–47487, 2024a.

---

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 6613–6623, 2024b.

Mingyuan Zhou, Huangjie Zheng, Yi Gu, Zhendong Wang, and Hai Huang. Adversarial score identity distillation: Rapidly surpassing the teacher in one step. arXiv preprint arXiv:2410.14919, 2024a.

Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In Forty-first International Conference on Machine Learning, 2024b.

Yuanzhi Zhu, Xingchao Liu, and Qiang Liu. Slimflow: Training smaller one-step diffusion models with rectified flow. In European Conference on Computer Vision, pp. 342–359. Springer, 2024.

---

## A DISCUSSION: WHY DOES CA WORK?

In this section, we attempt to build a conceptual bridge to understand the efficacy of CFG Augmentation (CA) as the "engine" of few-step distillation. We do so by drawing a parallel between diffusion models and Large Language Models (LLMs) under a unified view of sequential generation.

### A.1 A PARALLEL PROBLEM: WHY CAN'T LLMs PERFORM N-TOKEN PREDICTION?

We begin by considering a parallel question in the domain of LLMs: why must they generate text token by token, rather than predicting the next  $N$  tokens simultaneously? Consider the prompt, "The richest person in the world is". Both "Elon Musk" and "Bill Gates" are plausible completions. A model cannot simultaneously predict the first and second tokens, because the choice of the second token (e.g., "Musk" or "Gates") is strictly conditional on the choice of the first ("Elon" or "Bill"). An attempt to predict both at once would risk generating incoherent combinations like "Elon Gates" or "Bill Musk", or more likely, an averaged and meaningless representation.

The fundamental reason for this limitation is that the model's role is to predict a probability distribution for the next token,  $P(\text{token}_1 | \text{prompt})$ . It cannot, however, control the outcome of the probabilistic sampling process that selects a single token from this distribution. This external, uncontrollable sampling event breaks the model's predictive flow. No matter how powerful the model, it cannot bypass this external intervention to predict  $\text{token}_2$ , because any prediction it makes could conflict with the yet-undetermined outcome of  $\text{token}_1$ .

### A.2 A GENERAL FRAMEWORK FOR SEQUENTIAL GENERATION

We can abstract this to a higher level. Any sequential generator, at each step, faces three types of information:

- **Type 1 (Determined):** Information that is already fixed and known (e.g., the input prompt).
- **Type 2 (Directly Determinable):** Information for which the model can predict a distribution of possibilities for the very next step.
- **Type 3 (Directly Undeterminable):** Information that can only be determined after some Type 2 information is resolved and becomes Type 1.

The process of sequential generation is the iterative conversion of information from Type 2 to Type 1, which in turn allows Type 3 to become Type 2. Importantly, we claim that **the existence of Type 3 information, whose resolution is contingent on an uncontrollable external decision on Type 2 information, is the core reason for iterative generation**.

### A.3 FROM UNCONTROLLABLE INTERVENTION TO A DETERMINISTIC PATTERN

This analysis also points to a potential strategy for converting a sequential generator into a one-step generator: if the random, external decision-making process could be replaced with a **deterministic decision pattern**, and this pattern could be "baked into" the generator itself. For Type 2 information, what was previously a random variable with non-zero entropy would become a determinable value. This would collapse the entire decision tree into a single, predictable path, allowing all information to be resolved in one go.

### A.4 CONNECTING BACK TO DIFFUSION MODELS AND CFG AUGMENTATION

We posit that diffusion models are also sequential generators, which **first** establish low-frequency global composition (e.g., the object is a cat, not a dog) **before** adding high-frequency details (e.g., the texture of the fur). The relationship between composition and detail mirrors that of "Elon/Bill" and "Musk/Gates". We note that this viewpoint has been formally established [Dieleman \(2024\)](#).

Crucially, we argue that **Classifier-Free Guidance (CFG) acts as an external intervention analogous to probabilistic sampling**. While CFG is a deterministic bias, not a stochastic process, it is equally unpredictable from the model's perspective: The model is trained without awareness of CFG, and at inference, it cannot control the negative prompt or guidance scale ( $\alpha$ ) that will be applied.

---

Furthermore, CFG, like sampling, transforms the model’s prediction from an *averaged expectation* into a specific, shifted *value*.

Our central hypothesis is this: **CFG represents a specific, deterministic decision pattern.** The CA term in the DMD objective is the mechanism that "**bakes**" **this decision pattern into the student generator’s predictions.** By doing so, it transforms the uncontrollable external force of CFG into an internalized, predictable behavior. The generation process, which was a tree of possibilities, collapses into a single, direct path.

Returning to our LLM example, what CFG Augmentation does is akin to telling the model: "Given the current input, the external process will always choose 'Elon' as the first token. Therefore, you can safely assume the first token is 'Elon' and directly predict 'Musk'." This is, we believe, the source of CA’s power in enabling few-step image generation.

We acknowledge that the preceding discussion remains at the level of high-level ideas and that our hypothesis—that "CFG represents a specific, deterministic decision pattern"—is a strong assumption. We share this perspective here primarily to stimulate further investigation into this fundamental question and to provide a potential reference point for future work. We also intend to conduct a more in-depth study of this problem in our future research.

---

## B PSEUDO-CODE

---

**Algorithm 1** Original&Decoupled DMD Training Procedure
 

---

**Require:** Pre-trained teacher model  $s_{\text{real}}$ , CFG scale  $\alpha$ , number of steps  $N$ , proxy loss weight  $\lambda$

**Ensure:** Trained few-step generator  $G_\theta$

```

1: ▷ Initialize student generator and fake model from the teacher
2:  $G_\theta \leftarrow s_{\text{real}}$ 
3:  $s_{\text{fake}} \leftarrow s_{\text{real}}$ 
4: while not converged do
5:   ▷ — Generator Update Step —
6:   Sample a generation step  $t$  from the few-step schedule  $\{t_1, \dots, t_N\}$ 
7:   Prepare generator input  $z_t$  (e.g., via backward simulation for  $t > t_1$ )
8:   Generate an image:  $x_{\text{gen}} \leftarrow G_\theta(z_t)$ 
9:   if ‘decoupled_schedule’ then
10:    ▷ — Decoupled DMD behavior —
11:    Sample CFG augmentation noise level  $\tau_{\text{CA}} \sim \mathcal{U}(t, 1)$ 
12:    Sample Distribution Matching noise level  $\tau_{\text{DM}} \sim \mathcal{U}(0, 1)$ 
13:    Re-noise the generated image for both schedules:
14:     $x_{\tau_{\text{CA}}} \leftarrow \text{renoise}(x_{\text{gen}}, \tau_{\text{CA}})$ 
15:     $x_{\tau_{\text{DM}}} \leftarrow \text{renoise}(x_{\text{gen}}, \tau_{\text{DM}})$ 
16:    Withwith torch.no_grad():
17:      ▷ Calculate scores for the Distribution Matching (DM) term
18:       $s_{\text{cond, DM}}^{\text{real}} \leftarrow s_{\text{real}}(x_{\tau_{\text{DM}}}, \tau_{\text{DM}}, \text{text})$ 
19:       $s_{\text{cond, DM}}^{\text{fake}} \leftarrow s_{\text{fake}}(x_{\tau_{\text{DM}}}, \tau_{\text{DM}}, \text{text})$ 
20:      ▷ Calculate scores for the CFG Augmentation (CA) term
21:       $s_{\text{cond, CA}}^{\text{real}} \leftarrow s_{\text{real}}(x_{\tau_{\text{CA}}}, \tau_{\text{CA}}, \text{text})$ 
22:       $s_{\text{uncond, CA}}^{\text{real}} \leftarrow s_{\text{real}}(x_{\tau_{\text{CA}}}, \tau_{\text{CA}}, '')$ 
23:    EndWith
24:    ▷ Compute the two components of the update direction
25:     $\Delta_{\text{DM}} \leftarrow s_{\text{cond, DM}}^{\text{real}} - s_{\text{cond, DM}}^{\text{fake}}$ 
26:     $\Delta_{\text{CA}} \leftarrow (\alpha - 1) (s_{\text{cond, CA}}^{\text{real}} - s_{\text{uncond, CA}}^{\text{real}})$ 
27:     $\Delta_{\text{total}} \leftarrow \Delta_{\text{DM}} + \Delta_{\text{CA}}$ 
28:  else
29:    ▷ — Original DMD behavior —
30:    Sample a single noise level  $\tau \sim \mathcal{U}(0, 1)$ 
31:    Re-noise the generated image:  $x_\tau \leftarrow \text{renoise}(x_{\text{gen}}, \tau)$ 
32:    Withwith torch.no_grad():
33:       $s_{\text{cond}}^{\text{real}} \leftarrow s_{\text{real}}(x_\tau, \tau, \text{text})$ 
34:       $s_{\text{uncond}}^{\text{real}} \leftarrow s_{\text{real}}(x_\tau, \tau, '')$ 
35:       $s_{\text{cond}}^{\text{fake}} \leftarrow s_{\text{fake}}(x_\tau, \tau, \text{text})$ 
36:       $s_{\text{cfg}}^{\text{real}} \leftarrow s_{\text{uncond}}^{\text{real}} + \alpha(s_{\text{cond}}^{\text{real}} - s_{\text{uncond}}^{\text{real}})$ 
37:    EndWith
38:    ▷ Compute the combined update direction
39:     $\Delta_{\text{total}} \leftarrow s_{\text{cfg}}^{\text{real}} - s_{\text{cond}}^{\text{real}}$ 
40:  end if
41:  ▷ Update generator by minimizing the proxy loss
42:   $\mathcal{L}_{\text{proxy}} \leftarrow \|G_\theta(z_t) - \text{stop\_grad}(G_\theta(z_t) + \lambda \Delta_{\text{total}})\|^2$ 
43:  Update  $G_\theta$  by minimizing  $\mathcal{L}_{\text{proxy}}$ 
44:
45:  ▷ — Fake Model Update Step —
46:  ▷ This step can be run multiple times per generator update (TTUR)
47:  Sample a new noise level  $\tau' \sim \mathcal{U}(0, 1)$ 
48:  Generate a new image with detached gradient:  $x'_{\text{gen}} \leftarrow \text{stop\_grad}(G_\theta(z_t))$ 
49:  Re-noise the new image:  $x'_{\tau'} \leftarrow \text{renoise}(x'_{\text{gen}}, \tau')$ 
50:   $\mathcal{L}_{\text{denoise}} \leftarrow \|s_{\text{fake}}(x'_{\tau'}, \tau') - x'_{\text{gen}}\|^2$ 
51:  Update  $s_{\text{fake}}$  using  $\nabla \mathcal{L}_{\text{denoise}}$ 
52: end while

```

---

## C USER STUDY

To further validate the effectiveness of our proposed Decoupled-Hybrid schedule (④), we conducted a comprehensive user study comparing the four few-step models from the ablation in Table 1. The study was divided into two parts: a per-image ranking evaluation and a per-model side-by-side comparison.

### C.1 PER-IMAGE RANKING EVALUATION

In this part, we compared the visual quality of models ② (Decoupled-Full), ③ (Decoupled-Constrained), and ④ (Decoupled-Hybrid). We randomly selected 500 prompts from the HPSv2 benchmark Wu et al. (2023) and generated one image for each prompt using the three models. We then invited 10 professional annotators to perform a forced ranking of the three images in each set based on their overall quality (e.g., a ranking of 2>3>1, with ties disallowed). To prevent positional bias, the order of the three images within each triplet was randomized, so annotators could not determine which model generated which image based on its position.

The quantitative results are presented in Table 3 and Table 4. As shown, our proposed Decoupled-Hybrid schedule (④) demonstrates a significant and consistent advantage over the other configurations. It achieved a first-place ranking in **59.8%** of the evaluations, far surpassing the 33.8% of the next-best model, ③. The pairwise comparison in Table 4 further confirms this superiority, where model ④ achieved win rates of 60.6% and 83.4% against models ③ and ②, respectively.

Table 3: Overall performance and rank distribution from the per-image user study. Our Decoupled-Hybrid schedule (④) was ranked first in the majority of evaluations.

Model	Avg. Rank ↓	1st Place (%) ↑	2nd Place (%)	3rd Place (%)
② Decoupled-Full	2.748	6.4	12.4	<b>81.2</b>
③ Decoupled-Constrained	1.692	33.8	63.2	3.0
④ Decoupled-Hybrid	<b>1.560</b>	<b>59.8</b>	24.4	15.8

Table 4: Pairwise win rates (%) from the per-image ranking evaluation. Each cell (row, col) shows the percentage of times the model in the row was ranked higher than the model in the column.

	② Decoupled-Full	③ Decoupled-Constrained	④ Decoupled-Hybrid
② Decoupled-Full	—	8.6	16.6
③ Decoupled-Constrained	91.4	—	39.4
④ Decoupled-Hybrid	<b>83.4</b>	<b>60.6</b>	—

### C.2 PER-MODEL SIDE-BY-SIDE COMPARISON

We further conducted a per-model comparison to gauge the overall preference between different schedules. In this setup, we performed three separate side-by-side comparisons: ① (Coupled-Shared) vs. ④, ② (Decoupled-Full) vs. ④, and ③ (Decoupled-Constrained) vs. ④. For each comparison, we randomly sampled 200 prompts from the HPSv2 benchmark (using a different random seed for each pair) and displayed the generated images in a fixed two-column layout. We then asked 15 annotators to review all 200 image pairs and make a single, model-level judgment on which model (left or right) performed better overall. They were also asked to provide a brief justification for their choice.

Table 5: Quantitative results from the per-model, side-by-side user study with 15 annotators. Our Decoupled-Hybrid model (④) achieved a unanimous 100% preference rate in all comparisons.

Comparison	# Prompts	Preference for Model ④ (%)
① (Coupled-Shared) vs. ④	200	<b>100%</b>
② (Decoupled-Full) vs. ④	200	<b>100%</b>
③ (Decoupled-Constrained) vs. ④	200	<b>100%</b>

---

The quantitative results, summarized in Table 5, show a decisive victory for our proposed Decoupled-Hybrid model (④). It was unanimously preferred in all three head-to-head comparisons, achieving a **100%** win rate. The justifications provided by the annotators shed light on the reasons for this strong preference. The most frequently cited advantages of our model (④) were its ability to generate **richer details**, produce a more **realistic/less over-saturated/not greasy** texture/coloring, and exhibit **fewer anatomical or structural deformities**.

---

## D ADDITIONAL EXPERIMENTAL RESULTS

### D.1 DETAILED RESULTS OF RE-NOISING SCHEDULE ABLATION

In Tab. 1 of the main text, we have already presented the overall performance comparison for different re-noising schedule configurations. In Tab. 6 and Tab. 7, we additionally provide the fine-grained results on the HPS v2.1 and HPS v3 benchmarks, respectively.

Table 6: Detailed results of the Tab. 1 experiment on the HPS v2.1 benchmark.

Method	Concept-Art	Photo	Anime	Paintings	Average
Original (50 steps)	30.35	28.24	31.74	30.47	30.20
① $\tau_{CA} = \tau_{DM} \in [0, 1]$ (DMD)	30.31	29.95	31.72	30.45	30.61
② $\tau_{CA} \in [0, 1], \tau_{DM} \in [0, 1]$	30.31	30.25	31.85	30.34	30.69
③ $\tau_{CA} > t, \tau_{DM} > t$	31.48	<b>30.87</b>	32.98	31.51	31.71
④ $\tau_{CA} > t, \tau_{DM} \in [0, 1]$	<b>32.37</b>	<b>30.87</b>	<b>33.61</b>	<b>32.31</b>	<b>32.29</b>

Table 7: Detailed results of the Tab. 1 experiment on the HPS v3 benchmark.

Method	Animals	Architecture	Arts	Characters	Design	Food
Original (50 steps)	9.562	10.418	9.292	10.822	8.358	10.160
① $\tau_{CA} = \tau_{DM} \in [0, 1]$ (DMD)	10.377	11.537	9.077	11.061	8.112	11.520
② $\tau_{CA} \in [0, 1], \tau_{DM} \in [0, 1]$	10.338	11.539	8.929	11.101	8.059	11.572
③ $\tau_{CA} > t, \tau_{DM} > t$	11.309	12.388	9.690	11.975	8.694	12.096
④ $\tau_{CA} > t, \tau_{DM} \in [0, 1]$	11.873	12.899	10.351	12.562	9.308	12.425

Method	Nat. Sce.	Others	Plants	Products	Science	Transportation
Original (50 steps)	9.781	9.623	9.881	9.409	8.477	9.659
① $\tau_{CA} = \tau_{DM} \in [0, 1]$ (DMD)	10.138	10.797	10.528	10.994	8.417	11.503
② $\tau_{CA} \in [0, 1], \tau_{DM} \in [0, 1]$	10.046	10.787	10.600	11.049	8.358	11.484
③ $\tau_{CA} > t, \tau_{DM} > t$	11.177	11.553	11.390	11.508	8.905	12.256
④ $\tau_{CA} > t, \tau_{DM} \in [0, 1]$	11.592	12.138	11.846	11.841	9.451	12.782

## D.2 QUALITATIVE COMPARISON OF DIFFERENT REGULARIZERS

In Fig. 3 of the main text, we have already traced the quantitative indicators of combining the CFG Augmentation (CA) with different regularizers. In Fig. 6, we additionally provide a visualization of the generated samples from this experiment.

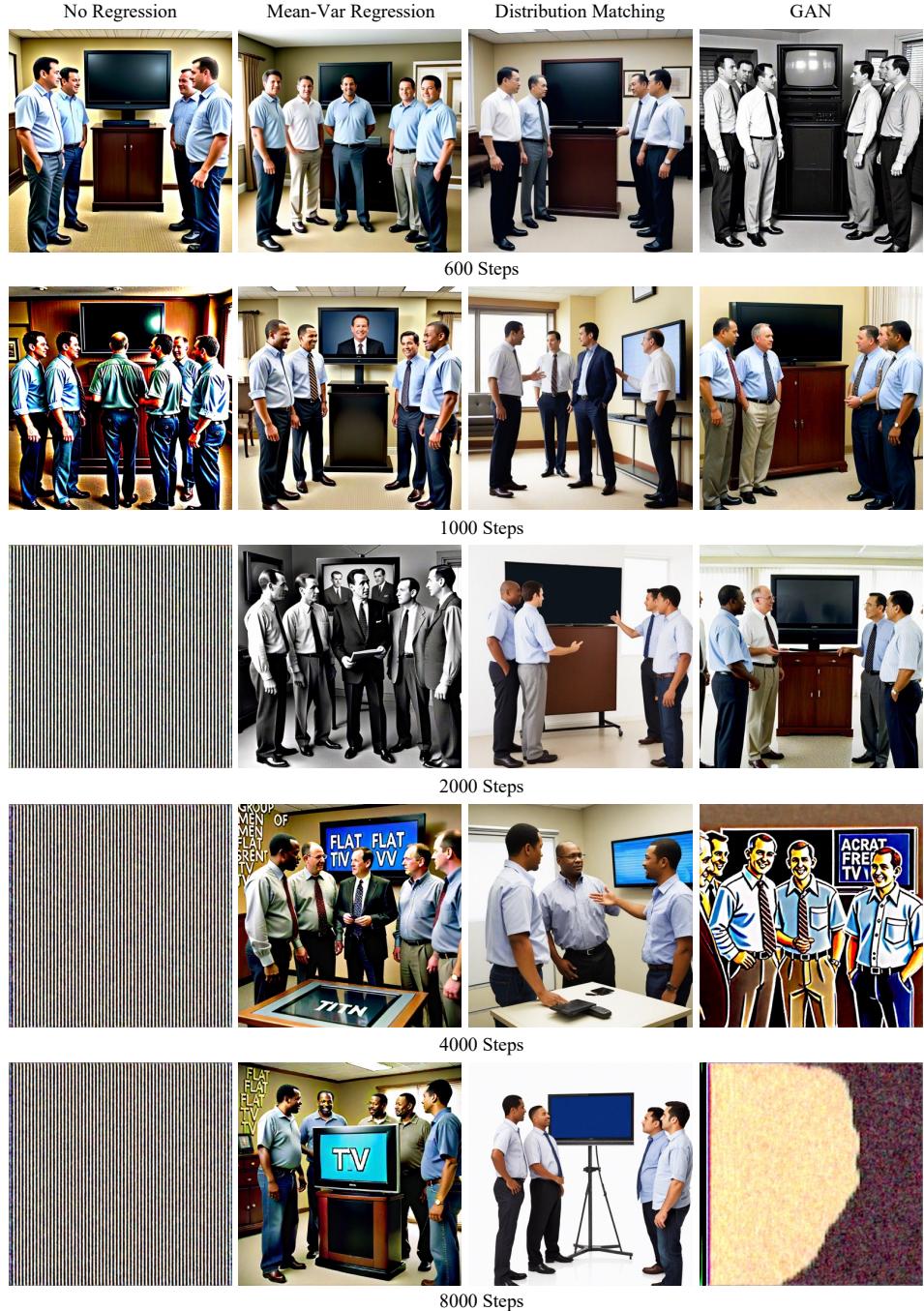


Figure 6: Sample visualization on combining training CA with different regularizers (Fig. 3)