

RL-NATIVE DISTILLATION: EXPLOITING SCORED TRAJECTORIES FOR FEW-STEP IMAGE GENERATION

Yuhan Li^{1,*} Fangao Zeng^{2,*,†} Sicong Kang^{2,*} Mengfei Xu² Hao Zhou²

Wenxiang Shang² Wei Li² Pipei Huang^{2,‡} Bingbing Ni^{1,‡}

*Equal Contribution †Project Leader ‡Joint Corresponding Authors

¹Shanghai Jiao Tong University ²Taobao & Tmall group of Alibaba

ABSTRACT

Efficient text-to-image generation requires both reinforcement-learning (RL)-based reward alignment and few-step distillation, yet these procedures are typically performed sequentially, increasing training cost and risking the loss of reward gains during compression. We instead take an *RL-native* perspective: diffusion RL already generates reward-scored finite-step trajectories, whose intermediate states provide a natural source of distillation supervision rather than a disposable byproduct of sampling. Based on this insight, we propose **REST** (Reward-Enhanced Scored-Trajectory Distillation), a single-stage RL-distillation co-training framework that attaches a decoupled student to an arbitrary RL teacher. The student learns segment-wise from the teacher’s evolving rollout trajectories while leaving the original teacher optimization unchanged. To prevent uniform imitation from preserving undesirable low-reward behaviors, we further introduce *Advantage-Modulated Distillation* (AMD), which transforms rollout advantages into signed weights over a base distillation loss. AMD strengthens supervision from preferred trajectories and mildly repels the student from low-reward ones. The resulting framework is lightweight and plug-and-play, requires no extra image rollouts, no separate distillation dataset, and no adversarial training. Experiments on compositional generation, visual text rendering, and human-preference alignment show that REST enables few-step CFG-free inference that matches or surpasses its 40-step RL teacher, with an overall additional training cost below **25%** over pure RL. REST improves DrawBench PickScore over RTDMD by 0.82 while requiring only **one-fifth** of the training iterations.

1 INTRODUCTION

Recent progress in diffusion and flow-matching models (forest labs, 2024; Wu et al., 2025) has substantially improved the visual quality of text-to-image generation. In real practice, however, two post-training procedures are often required before such models become truly useful (Jiang et al., 2025; Fan et al., 2026): few-step sampling and classifier-free guidance (CFG) (Ho & Salimans, 2022) distillation (Luo et al., 2023; Yin et al., 2024b) for efficient inference, and reinforcement-learning-based alignment for human preference (Shao et al., 2024). These two procedures are traditionally executed sequentially, which is cumbersome and often suboptimal: distillation may wash out the reward gains obtained from RL, while subsequent RL can disturb a previously distilled few-step generator and degrade its structure or low-step inference stability (Jiang et al., 2025).

A natural question is whether reward alignment and few-step distillation can be performed in a unified post-training stage. As in Tab. 1, recent works such as DMDR (Jiang et al., 2025) and RTDMD (Huang et al., 2026) make progress in this direction by introducing a warm-up distribution-matching distillation (Yin et al., 2024b) before the RL process, followed by RL+DMD parallel optimization. The multi-stage system often involves balancing a number of stage-specific hyperparameters as well as using more data and training iterations. They also suffer from complicated pipelines compared to RL algorithms, such as an extra fake model, extra rollouts, and alternating training. More importantly, the frozen RL-agnostic teacher distribution used by the distillation ob-

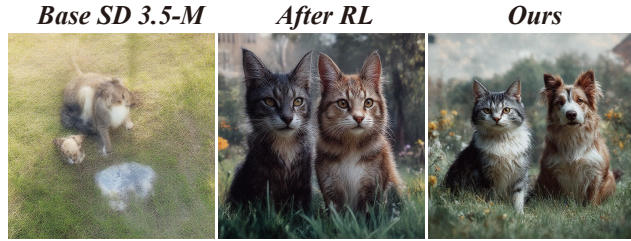
jective may pull against the continuously evolving reward-optimized policy, creating an inherent tension between imitation and reward improvement.

In this work, we take an *RL-native* perspective: distillation should be built directly into RL training, without a separate pre-distillation stage, external data, or auxiliary models such as fake-score networks. Instead, it should reuse the rollouts, rewards, and intermediate states already produced by RL and exploit RL’s own capacity for step reduction. We observe that diffusion RL already possesses an implicit capacity for step reduction, because the policy is directly optimized through terminal samples generated by a finite-step rollout (Fig. 1). Following this principle, we attach a decoupled student branch to an arbitrary diffusion RL algorithm. The teacher retains its original reward-optimization objective, while the student performs segment-wise imitation along the same reward-scored rollout trajectory without affecting teacher optimization. This turns standard RL sampling into native distillation supervision and improves few-step CFG-free inference. Nevertheless, uniform imitation may still preserve undesirable behaviors from low-reward rollouts, such as structural ambiguity and fine-grained artifacts.

To address this limitation, inspired by Xue et al. (2025a), we further reuse the rollout reward in student distillation through *Advantage-Modulated Distillation* (AMD). AMD applies an affine transformation to the reward advantage of each teacher rollout, yielding a signed modulation coefficient for the base distillation loss. This mechanism induces a contrastive effect analogous to CFG (Ho & Salimans, 2022) and NFT (Chen et al., 2026): high-reward trajectories exert stronger attractive supervision toward desirable teacher behaviors, whereas sufficiently low-reward trajectories provide a mild repulsive gradient that discourages the student from reproducing undesirable ones. Compared with uniform distillation, this signed reward modulation sharpens the few-step student and improves visual clarity. Because AMD only reweights the underlying per-sample objective without changing its target, it also serves as a general reward-aware wrapper for different distillation losses.

In summary, we present **REST** (Reward-Enhanced Scored-Trajectory Distillation), a unified RL-distillation co-training framework for efficient and reward-aligned diffusion generation. To the best of our knowledge, it is the first single-stage framework for RL-distillation collaboration. It has three key advantages. First, it delivers high-quality few-step generation, matching or even surpassing the full-step inference RL-trained teacher: the general AMD reward modulation compensates for the

Methods	Data	Stages	Teacher	Train Steps
DMDR	2M data	2	Frozen Base	1.5k+1.5k
RTDMD	2M data	2	Frozen Base	1.5k+1.0k
RL then Distill	25k RL data	2	After-RL	500+1.5k
Distill then RL	Both	2	Frozen Base	1.5k+1k
Ours	25k RL data	1	RL-evolving	500



Prompt: One cat and one dog sitting on the grass.



Prompt: A photo of a fire hydrant and a tennis racket.



A vibrant "Choco Crunch" candy wrapper with the bold brand name on a colorful background, featuring chocolate chunk illustrations.

Figure 1: REST achieves few-step, reward-aligned generation through a simple and efficient training process as in the table above. We also present 8-step CFG-free samples across the PickScore, OCR, and GenEval datasets in the figure below. While RL-trained models have demonstrated preliminary few-step generation capabilities, our approach delivers higher-quality outputs with superior preference alignment.

quality loss introduced by pure distillation. Second, it is plug-and-play: the decoupled teacher–student design and the generic AMD formulation make the method independent of a specific teacher RL algorithm or a particular distillation loss, which is present in Sec. 4.4. Third, it is simple and efficient: the framework avoids warm-up scheduling and delicate coordination among multiple competing objectives; it also requires no extra image rollouts, no separate distillation dataset, and no fake-score model training as in DMDR and RTDMD. Our contributions are summarized as follows:

- We propose REST, a decoupled teacher–student co-training framework that attaches a lightweight student branch to an arbitrary RL teacher branch. The student reuses the teacher’s existing rollout trajectory and reward scores, introducing little extra sampling cost while avoiding interference with teacher RL optimization.
- We introduce AMD, a general reward-aware modulation mechanism for arbitrary distillation losses. We show that the scale-and-shift transformation of advantages naturally introduces negative samples and a CFG-like contrastive effect, which improves the quality of the distilled few-step student.
- We conduct extensive experiments showing that, with only **26.1%** additional training overhead over the base RL pipeline, our method enables CFG-free few-step inference that approaches the quality of the full-inference RL teacher. Compared with a naive two-stage RL-then-distill pipeline or existing unified RL-distillation methods, it requires only 1/5 training iterations and obtains substantially better alignment performance.

2 RELATED WORK

Reinforcement learning for diffusion alignment. Reinforcement learning has become a major paradigm for aligning diffusion and flow-matching models with human preferences. Policy-gradient methods such as Flow-GRPO (Liu et al., 2025), DanceGRPO (Xue et al., 2025b), MixGRPO (Li et al., 2025), AWM (Xue et al., 2025a) and DiffusionNFT (Zheng et al., 2025) optimize terminal rewards over denoising trajectories, while preference-based methods (Rafailov et al., 2023; Wallace et al., 2024; Liang et al., 2025) avoid explicit reward modeling through pairwise supervision. Direct-gradient approaches such as ReFL (Xu et al., 2023), DRaFT (Clark et al., 2024), DRTune (Wu et al., 2024), and LeapAlign (Liang et al., 2026) propagate reward gradients through differentiable sampling paths, which require differentiable rewards. Despite their success in improving reward-aligned sampling, these methods largely overlook the potential of integrating distillation into RL, and thus the possibility of achieving reward-aligned few-step generation remains underexplored.

Few-step diffusion distillation. Few-step distillation aims to compress multi-step diffusion samplers into generators requiring only one or a few inference steps. Early trajectory-distillation methods directly regress the student toward the outputs of the teacher’s ODE trajectory: Knowledge Distillation (Luhman & Luhman, 2021) matches the teacher’s full sampling result in a single step, and Progressive Distillation (Salimans & Ho, 2022) iteratively halves the number of sampling steps by training the student to match two teacher steps at a time. While consistency-based methods, including Consistency Models (Song et al., 2023), Latent Consistency Models (Luo et al., 2023), and Phased Consistency Models (Wang et al., 2024), enforce trajectory-level consistency, score-based methods such as SwiftBrush (Nguyen & Tran, 2024) and Score Implicit Matching (Luo et al., 2024) pursue fast generation through score distillation. Distribution Matching Distillation (DMD) (Yin et al., 2024b) and DMD2 (Yin et al., 2024a) improve few-step quality by matching student and teacher distributions with auxiliary fake-score or discriminator-like networks. However, these distillation objectives are usually reward-agnostic and often require a frozen teacher, a separate distillation stage, or additional auxiliary models. REST is orthogonal to the choice of base distillation loss: it modulates common distillation losses with reward-derived advantages, making distillation preference-aware without redesigning the underlying distillation algorithm.

Unified RL-distillation training. Sequentially applying RL alignment and few-step distillation is cumbersome and can cause objective interference: distillation may wash out RL gains, while later RL can destabilize a previously distilled few-step generator. Recent works such as DMDR (Jiang et al., 2025) and RTDMD (Huang et al., 2026) integrate DMD-style distillation with RL, but their fake-score-centered designs involve multi-network training, warm-up, and staged optimization, and their frozen reference distributions may lag behind the continuously evolving reward-optimized policy. REST instead uses a decoupled dual-branch design: the teacher continues its original RL op-

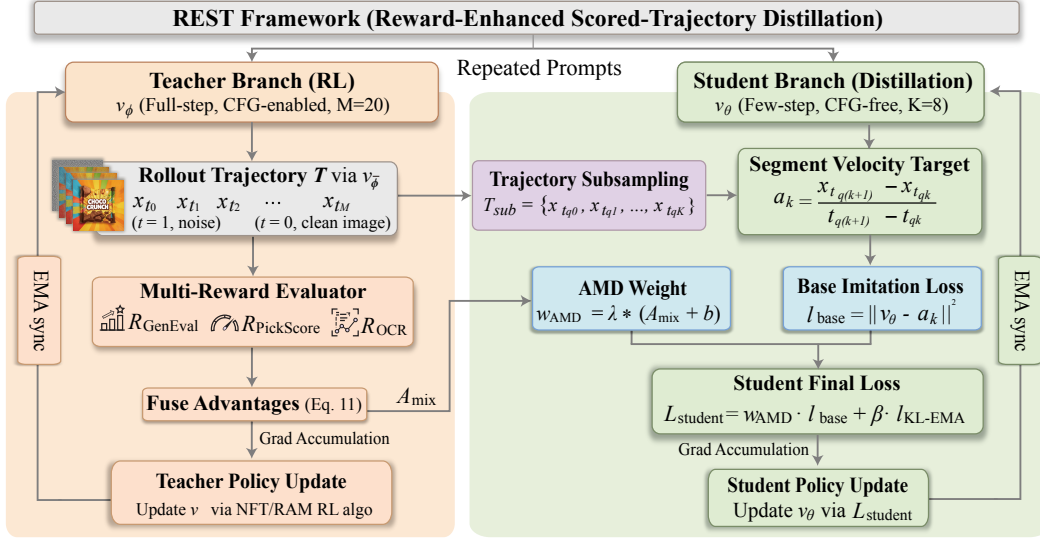


Figure 2: Overview of REST. The decoupled student reuses reward-scored teacher rollouts, while AMD modulates trajectory distillation without altering teacher RL optimization.

timization, while the student simultaneously learns a few-step policy from the same reward-scored teacher rollouts through REST. This yields a unified RL-distillation co-training framework that requires no extra image rollouts, no separate distillation dataset, and no fake-score model training.

3 METHOD

3.1 PRELIMINARY: FORWARD-PROCESS DIFFUSION RL

Our teacher branch can in principle be optimized by any diffusion RL algorithm, but it pairs especially well with forward-process RL algorithms. The representative instances include Diffusion-NFT (Zheng et al., 2025), AWM (Xue et al., 2025a) and RAM (Bergmeister et al., 2026), which all cast reward alignment as a reward-weighted velocity regression on the forward noising process $x_t = (1 - t)x_0 + t\epsilon$, where x_0 is the rollout sample and $\epsilon \sim \mathcal{N}(0, I)$. DiffusionNFT converts a scalar reward into an optimality probability $r \in [0, 1]$ and implicitly parameterizes a positive and a negative policy around the old velocity v^{old} ,

$$v_\phi^+ = (1 - \beta)v^{\text{old}} + \beta v_\phi(x_t), \quad v_\phi^- = (1 + \beta)v^{\text{old}} - \beta v_\phi(x_t), \quad (1)$$

and trains the policy v_ϕ with

$$\mathcal{L}_{\text{NFT}}(\phi) = \mathbb{E} \left[r \left\| v_\phi^+(x_t) - v^{gt} \right\|^2 + (1 - r) \left\| v_\phi^-(x_t) - v^{gt} \right\|^2 \right], \quad (2)$$

where $v^{gt} = \epsilon - x_0$ is the forward flow-matching (Lipman et al., 2022) target. RAM instead directly regresses v_ϕ toward a reward-shifted version of the reference velocity,

$$\mathcal{L}_{\text{RAM}}(\phi) = \mathbb{E}_t \left[\left\| v_\phi(x_t) - \text{sg}(v^{\text{base}}(x_t) + r(x_0)(v^{gt} - v_\phi(x_t))) \right\|^2 \right], \quad (3)$$

where $\text{sg}(\cdot)$ is the stop-gradient operator and $v^{\text{base}}(\cdot)$ means frozen base model. In our framework, the teacher branch is optimized by any forward-process algorithm, e.g., Eq. 2 or Eq. 3, producing an evolving teacher policy v_ϕ together with reward-scored rollouts that the student branch reuses.

3.2 DECOUPLED TEACHER-STUDENT CO-TRAINING

Given the teacher policy v_ϕ as in 3.1, REST attaches a decoupled student branch v_θ that learns a few-step, CFG-free generator from the teacher’s rollouts, without altering the teacher optimization,

as shown in Fig. 2. For each prompt c , the teacher branch samples with its ODE solver (Song et al., 2020) on an M -step schedule (e.g., $M = 20$) with classifier-free guidance (CFG), producing a trajectory

$$\mathcal{T} = \{x_{t_0}, x_{t_1}, \dots, x_{t_M}\}, \quad 1 = t_0 > t_1 > \dots > t_M = 0, \quad (4)$$

and a terminal reward $r = R(x_0, c)$. The student, in contrast, is designed to run with only $K \ll M$ steps (e.g., $K = 8$) at inference time. Rather than sampling a separate trajectory for the student, we select a K -step subset of the teacher’s schedule and re-index it as an imitation trajectory

$$\mathcal{T}_{\text{sub}} = \{x_{t_{q0}}, x_{t_{q1}}, \dots, x_{t_{qK}}\} \subset \mathcal{T}, \quad \{t_{q0}, \dots, t_{qK}\} \subset \{t_0, \dots, t_M\}, \quad (5)$$

so that the student can be trained directly on the teacher’s existing rollout, with no additional sampling required. For each student step k , the *piecewise trajectory velocity* of the corresponding segment is directly computed from the teacher’s rollout,

$$v_k^{\text{gt}} = \frac{x_{t_{q(k+1)}} - x_{t_{qk}}}{t_{q(k+1)} - t_{qk}}. \quad (6)$$

In this decoupled co-training framework, the student model θ is trained to imitate this piecewise trajectory, as detailed in Sec. 3.3, thereby inheriting the teacher’s generation quality while enabling few-step CFG-free sampling.

This dual-branch design brings two benefits. First, it is a one-stage, end-to-end training framework: the student is trained by reusing the teacher’s existing rollout trajectory, making training more efficient than a two-stage pipeline. Second, because the teacher policy v_ϕ continues to evolve under RL training, avoiding a frozen teacher that could hold back the student branch’s RL progress.

3.3 ADVANTAGE-WEIGHTED REGRESSION FOR TEACHER TRACKING

Advantage-weighted regression (Peters & Schaal, 2007; Peng et al., 2019; Kostrikov et al., 2021), *i.e.* AWR, recasts policy improvement as a reward-weighted supervised regression problem: it treats each observed state-action pair as a demonstration and re-weights its log-likelihood under the current policy by the advantage of that action, so that high-advantage actions are imitated more strongly while low- or negative-advantage actions are discouraged. We adopt this view to let the student track the teacher’s reward-scored rollout: each teacher trajectory segment is treated as a state-action demonstration, whose imitation strength is modulated by the reward of the rollout it comes from. For prompt condition c , the state-action pair $s_k = (x_{t_{qk}}, t_{qk}, c)$ and the demonstrated action $a_k = v_k^{\text{gt}}$ given by the teacher’s segment velocity (Eq. 6). We interpret the distribution of student output as the mean of an implicit, fixed-variance Gaussian policy following Liu et al. (2025) over teacher segment velocities,

$$\pi_\theta(a_k | s_k) = \mathcal{N}(a_k; v_\theta(s_k), \sigma^2 I), \quad (7)$$

under which the negative log-likelihood of the demonstrated action is, up to an additive constant independent of θ ,

$$-\log \pi_\theta(a_k | s_k) = \frac{1}{2\sigma^2} \|a_k - v_\theta(s_k)\|^2 + \text{const.} \quad (8)$$

Dropping the constant, this yields the plain *per-step imitation loss*

$$\ell_{\text{base}}(\theta; k) = \|v_\theta(s_k) - a_k\|^2 \propto -\log \pi_\theta(a_k | s_k), \quad (9)$$

which treats every teacher trajectory segment as equally reliable supervision. AWR improves a policy by imitating demonstrated actions with weights given by their advantages,

$$\max_{\theta} \mathbb{E}_{(s,a) \sim \mathcal{D}} [A(s, a) \log \pi_\theta(a | s)]. \quad (10)$$

Using Eq. 8, this objective induces an advantage-weighted regression loss over teacher trajectory segments, where the log-likelihood term is replaced by the base imitation loss ℓ_{base} . where $\mathcal{D}$ is the reward-scored teacher trajectory dataset formed by all segments (s_k, a_k) . For each reward source i , its advantage $A^{(i)}(s_k, a_k) \in [-1, 1]$ is the clipped group-normalized advantage of the rollout associated with prompt c , and is shared by all segments from the same rollout, as in Flow-GRPO (Liu et al., 2025) and AWM (Xue et al., 2025a). We further fuse different rewards on the normalized advantages level,

$$A_{\text{mix}} = \frac{\sum_{i=1}^{N_R} \alpha_i A^{(i)}}{\sum_{i=1}^{N_R} \alpha_i}, \quad \alpha_i \geq 0, \quad (11)$$

where N_R is the number of reward sources and α_i controls their contribution. We then introduce a global scale $\lambda = 1$ and a positive shift $b = 0.5$ to obtain a signed modulation coefficient, leading to our Advantage-Modulated Distillation (AMD) objective for teacher trajectory tracking:

$$\mathcal{L}_{\text{AMD}}(\theta) = \mathbb{E}_{(s_k, a_k) \sim \mathcal{D}} [\lambda (A_{\text{mix}} + b) \ell_{\text{base}}(\theta; k)]. \quad (12)$$

In practice, we further regularize the student against an EMA copy of itself, denoted by $v_{\theta_{\text{ema}}}$, using a fixed-variance Gaussian KL surrogate

$$\mathcal{L}_{\text{KL-EMA}}(\theta) = \mathbb{E}_k [\|v_{\theta}(s_k) - v_{\theta_{\text{ema}}}(s_k)\|^2], \quad (13)$$

which stabilizes optimization and reduces the training jitter caused by the continuously changing teacher policy. The final student objective is therefore

$$\mathcal{L}_{\text{student}}(\theta) = \mathcal{L}_{\text{AMD}}(\theta) + \beta \mathcal{L}_{\text{KL-EMA}}(\theta), \quad (14)$$

where β controls the strength of the EMA-student regularization.

3.4 UNDERSTANDING ADVANTAGE-MODULATED DISTILLATION

Decomposing the AMD objective. Equation 12 can be decomposed into two additive terms,

$$\mathcal{L}_{\text{AMD}}(\theta) = \underbrace{\lambda b \mathbb{E}_{(s_k, a_k) \sim \mathcal{D}} [\ell_{\text{base}}(\theta; k)]}_{\text{imitation prior}} + \underbrace{\lambda \mathbb{E}_{(s_k, a_k) \sim \mathcal{D}} [A_{\text{mix}} \ell_{\text{base}}(\theta; k)]}_{\text{reward-driven correction}}. \quad (15)$$

The first term is a constant-weighted imitation prior. A positive shift b is crucial in our setting: the initial student branch does not yet possess reliable image-generation capability, and therefore needs a stable positive imitation signal to bootstrap its few-step generator. From another perspective, even trajectories with relatively low advantages still contain useful teacher dynamics for the student, especially at early training stages, as shown in Fig. 6. The second term is the reward-driven correction inherited from reward-weighted supervised regression: it biases the student toward teacher segments with high combined reward advantages, allowing the few-step student to concentrate on the best parts of the teacher’s reward-scored rollouts and potentially surpass the average behavior of the RL-trained full-step teacher. This reward-aware distillation effect is unavailable to conventional trajectory distillation methods (Salimans & Ho, 2022) that imitate teacher rollouts uniformly.

AMD as a generic reward-aware distillation wrapper. Under the AWR view above, the base per-step imitation loss ℓ_{base} is instantiated as the segment-velocity MSE in Eq. 9. This MSE could be understood as the simplest trajectory distillation objective (Salimans & Ho, 2022) rather than a required design choice. Since AMD only modulates the weight in front of ℓ_{base} , the same mechanism can wrap more advanced distillation losses, such as PCM-style phase consistency losses (Wang et al., 2024), DMD-style distribution-matching losses (Yin et al., 2024a), or other trajectory-level objectives. Therefore, AMD can serve as a generic reward-aware wrapper over diffusion distillation objectives, making different student distillation algorithms benefit from the reward-scored teacher rollouts without changing the teacher’s RL training procedure.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks and rewards. Following the experimental protocol of RAM and Flow-GRPO (Liu et al., 2025), we consider three representative text-to-image reward objectives: compositional correctness (GenEval), visual text rendering (OCR), and human preference alignment (PickScore). GenEval (Ghosh et al., 2023) measures whether generated images satisfy object, attribute, counting, and spatial-relation constraints. OCR evaluates visual text rendering through an edit-distance reward that checks whether the target text specified in the prompt appears legibly in the image. PickScore (Kirstain et al., 2023) is a learned human-preference model trained from large-scale pairwise image comparisons.

For PickScore alignment, we use PickScore as the training reward. For OCR and GenEval, however, we use a multi-reward setting: OCR+PickScore and GenEval+PickScore, respectively. This design

is motivated by **an empirical failure mode of prior single-reward optimization** (Fig. 5): although OCR or GenEval scores can become high, *the generated images may collapse into reward-hacking artifacts, such as overly large text covering the whole image or simplified layouts that discard most visual details*. This collapse is reflected by DrawBench **quality metrics becoming even lower than the base model SD3.5M**. Adding PickScore as an auxiliary reward makes the optimization more conservative and better preserves general image quality while still improving the target task reward.

Training protocol and AMD settings. We use Stable Diffusion 3.5 Medium (SD3.5M) (Esser et al., 2024) as the backbone and train one LoRA (Hu et al., 2022) ($r = 32$, $\alpha = 64$) per reward setting with bf16 precision, learning rate 3×10^{-4} , 48 prompts per group, and 24 samples per prompt. PickScore models are trained for 500 steps, while GenEval and OCR models are trained for 300 steps. The teacher follows RAM’s forward-process RL setup: 20-step training solver, 40-step evaluation solver, and CFG-enabled sampling. The student is jointly trained with AMD and evaluated using the EMA student (decay 0.9) with 8 CFG-free steps. For AMD, each reward source uses group-relative advantages normalized within same-prompt samples to $[-1, 1]$; OCR/GenEval task advantages are fused with PickScore advantages via Eq. 11. We set the AMD scale to $\lambda = 1$, shift to $b = 0.5$, and KL-EMA coefficient to 0.2. Training rewards are evaluated on held-out benchmark prompts, while generic image quality is evaluated on DrawBench (Saharia et al., 2022).

Generic image-quality evaluation. Optimizing a certain reward can degrade generic image quality, a failure mode often referred to as reward hacking. Following DiffusionNFT (Zheng et al., 2025), we evaluate each trained model on DrawBench (Saharia et al., 2022) prompts that are disjoint from the reward-training and reward-test prompts. We report five quality and preference metrics: Aesthetic (Schuhmann & Beaumont, 2022) and DeQA (You et al., 2025) for perceptual quality, and ImageReward (Xu et al., 2023), HPSv2 (Wu et al., 2023), and PickScore (Kirstain et al., 2023) for human preference.

4.2 MAIN RESULTS

The full-step RL teacher provides the reward-aligned trajectory distribution that REST reuses for student training, and the REST student is designed to retain the gains of the 40-step RAM teacher while reducing inference to a few CFG-free steps. Tab. 1 summarizes the reward and image-quality results. For GenEval and OCR, the gray-shaded rows report single-reward baselines as diagnostic references: as discussed above, these settings can achieve high task rewards while severely degrading generic image quality. We therefore focus on the rows marked by $\dagger$, which use the multi-reward setting with the task reward and PickScore.

As shown in Tab. 1, REST achieves performance comparable to the 40-step RAM teacher on both training rewards and out-of-domain DrawBench metrics, while using only a few CFG-free inference steps. It also obviously surpasses full-step RL baselines such as Flow-GRPO (Liu et al., 2025), AWM (Xue et al., 2025a) and DiffusionNFT (Zheng et al., 2025), and substantially improves over the naive 8-step RAM (Bergmeister et al., 2026). We note that 8-step RAM can occasionally obtain a high Aesthetic score, but this is often associated with fragmented structures and noisy details that hack the metric rather than reflect better perceptual quality. Overall, by continuously exposing the student to the evolving reward-optimized teacher rollouts and using AMD to select and amplify high-value trajectory segments, REST enables the student to go beyond plain trajectory compression. Qualitative comparisons in Figs. 3, 4, and 5 further show that REST preserves generic image quality, human preference, and visual text rendering under few-step CFG-free sampling.

Compared with Few-step RL Methods. DMDR and RTDMD (Jiang et al., 2025; Huang et al., 2026) combine RL and DMD-style distillation through a DMD warm-up stage followed by parallel GRPO-DMD training. However, they use larger training datasets (Schuhmann et al., 2021) and longer training of roughly 2,500–3,000 steps. More importantly, neither work directly compares its distilled generator against the corresponding full-step RL model trained under the same setting, leaving unclear how much of the RL policy’s performance is actually preserved after distillation. REST instead targets the quality of full-step RL models under the popular GenEval, OCR, PickScore, and DrawBench protocols in diffusion RL fields, while requiring far fewer training steps. For an additional comparison, we evaluate the official RTDMD weights and reproduced RTDMD using official code on the fair rewards and prompts, at the training-independent DrawBench benchmark.

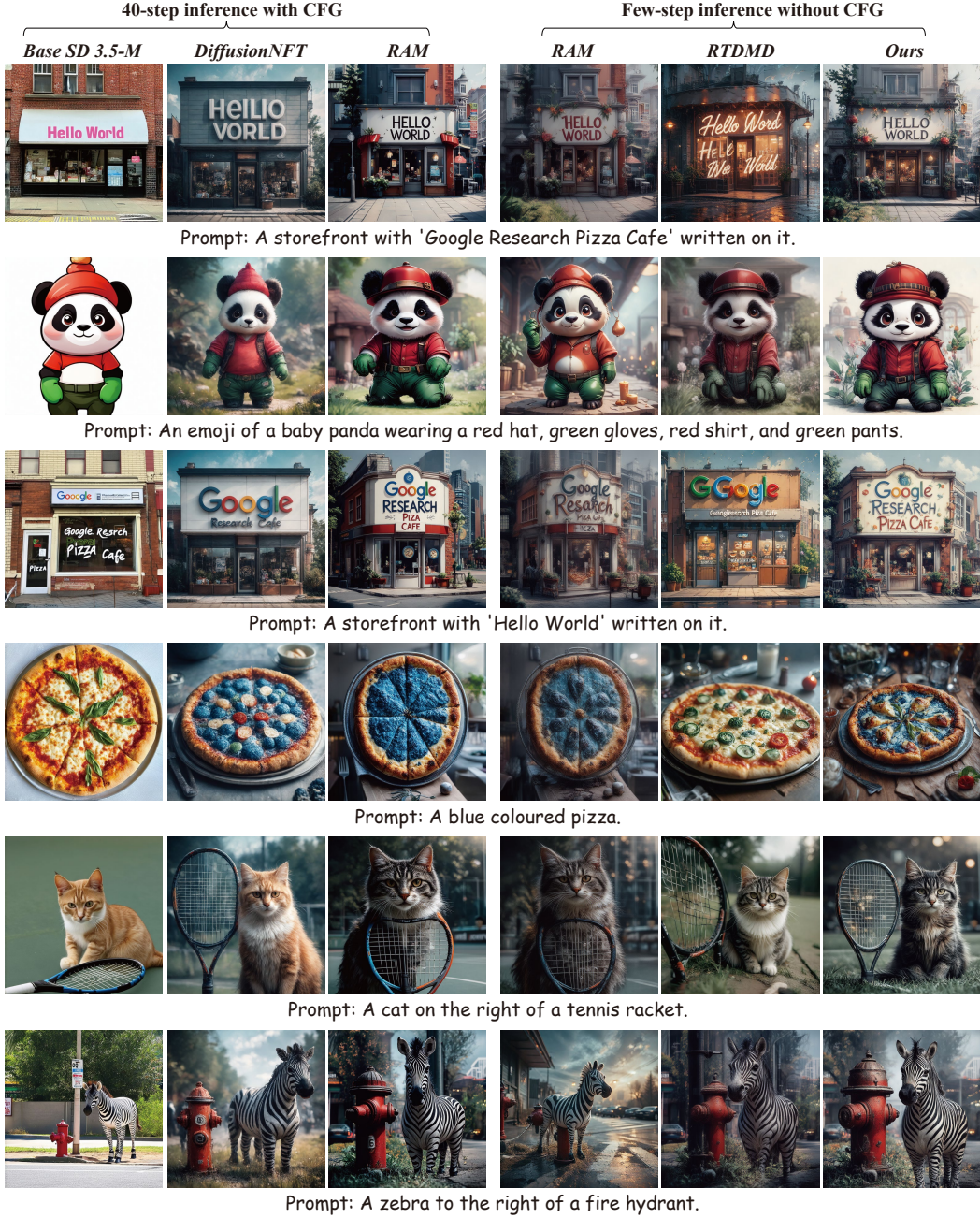


Figure 3: Qualitative comparison on DrawBench. REST achieves high-quality outputs with superior preference alignment under few-step inference, while matching full-step inference RL teacher.

As shown in Tab. 2, REST achieves stronger aesthetic and preference scores over both RTDMD versions despite using substantially fewer training steps.

4.3 ABLATION STUDY

Tab. 3 studies the contribution of each component under PickScore training. We start from direct 8-step CFG-free inference with the base model, then add standard RL (RAM), a sequential RL-then-distill two-stage pipeline, simultaneous dual-branch RL with distillation, and finally the full REST

Table 1: SD3.5M post-training results on three tasks (GenEval, OCR, PickScore). Gray rows use a single task reward; † denotes joint training with the task reward and PickScore. Higher is better.

Model	NFE	Training Reward			Image Quality Metrics				
		GenEval	OCR	PickScore	Aesthetic	DeQA	ImgRwd	HPSv2	PickScore
SD3.5M	40	0.64	0.63	21.78	5.39	4.08	0.85	0.28	22.40
SD3.5M	8	0.28	0.12	19.50	5.15	2.38	-0.96	0.18	20.39
<i>Compositional Image Generation</i>									
Flow-GRPO	40	0.95			5.25	4.01	1.03	0.27	22.37
AWM	40	0.83			5.14	3.75	0.67	0.24	22.04
DiffusionNFT	40	0.95			4.98	4.10	0.30	0.24	21.59
RAM	40	0.97			5.38	4.09	1.19	0.29	22.52
RAM†	40	0.95			5.42	4.12	1.12	0.28	22.76
RAM†	8	0.65			5.46	2.27	-0.19	0.20	21.36
Ours†	8	0.94			5.67	4.15	1.05	0.30	22.79
<i>Visual Text Rendering</i>									
Flow-GRPO	40		0.92		5.32	4.06	0.95	0.28	22.44
AWM	40		0.97		5.01	2.83	-0.85	0.18	20.56
DiffusionNFT	40		0.96		4.87	3.01	-0.97	0.18	20.26
RAM	40		0.97		5.23	3.90	0.44	0.26	21.83
RAM†	40		0.96		5.58	4.16	1.07	0.30	22.81
RAM†	8		0.68		5.59	2.60	-0.21	0.22	21.45
Ours†	8		0.96		5.65	4.16	1.00	0.30	22.83
<i>Human Preference Alignment</i>									
Flow-GRPO	40			23.31	5.92	4.22	1.28	0.32	23.53
AWM	40			23.39	6.31	4.10	1.27	0.31	23.76
DiffusionNFT	40			23.29	6.16	4.13	1.23	0.31	23.65
RAM	40			23.70	6.17	4.21	1.35	0.32	23.95
RAM	8			23.30	6.42	3.82	1.11	0.31	23.55
Ours	8			23.73	6.30	4.15	1.35	0.33	23.96

Table 2: Few-step DrawBench comparison with RTDMD. RTDMD training iterations include 1,500 warm-up and 1,000 parallel training iterations; † denotes officially released weights and ‡ denotes our reproduced version using official code on PickScore reward and Pick-a-Pic dataset.

Methods	NFE	Train Iters	Aesthetic	ImgRwd	HPSv2	PickScore	CLIPScore
RTDMD†	4	1500+1000	6.09	1.26	0.326	23.30	0.930
RTDMD†	8	1500+1000	6.27	1.20	0.330	23.14	0.916
RTDMD‡	4	1500+1000	6.18	1.27	0.310	23.71	0.885
RTDMD‡	8	1500+1000	6.07	1.21	0.310	23.49	0.886
REST (full)	8	500	6.30	1.35	0.333	23.96	0.949

model with AMD. The results lead to three observations. First, full-step standard RL itself improves 8-step CFG-free sampling, showing that reward optimization can partially compensate for the low-step generation gap. Second, simultaneous dual-branch RL-distillation in REST retains most of the gains of the RL-then-distill pipeline while reducing training cost by only 1/4 training iterations, since the student reuses the teacher rollouts instead of requiring a separate distillation stage. Third, AMD further strengthens reward preference and improves perceptual quality, indicating that reward-aware trajectory weighting is more effective than uniform imitation.

Tab. 4 further isolates the stabilizing components of the full AMD objective in the same PickScore setting. The shift b controls the positive imitation prior in the AMD weight. Setting $b = 0$ removes this baseline imitation pressure, making the objective dominated by aggressive negative modulation

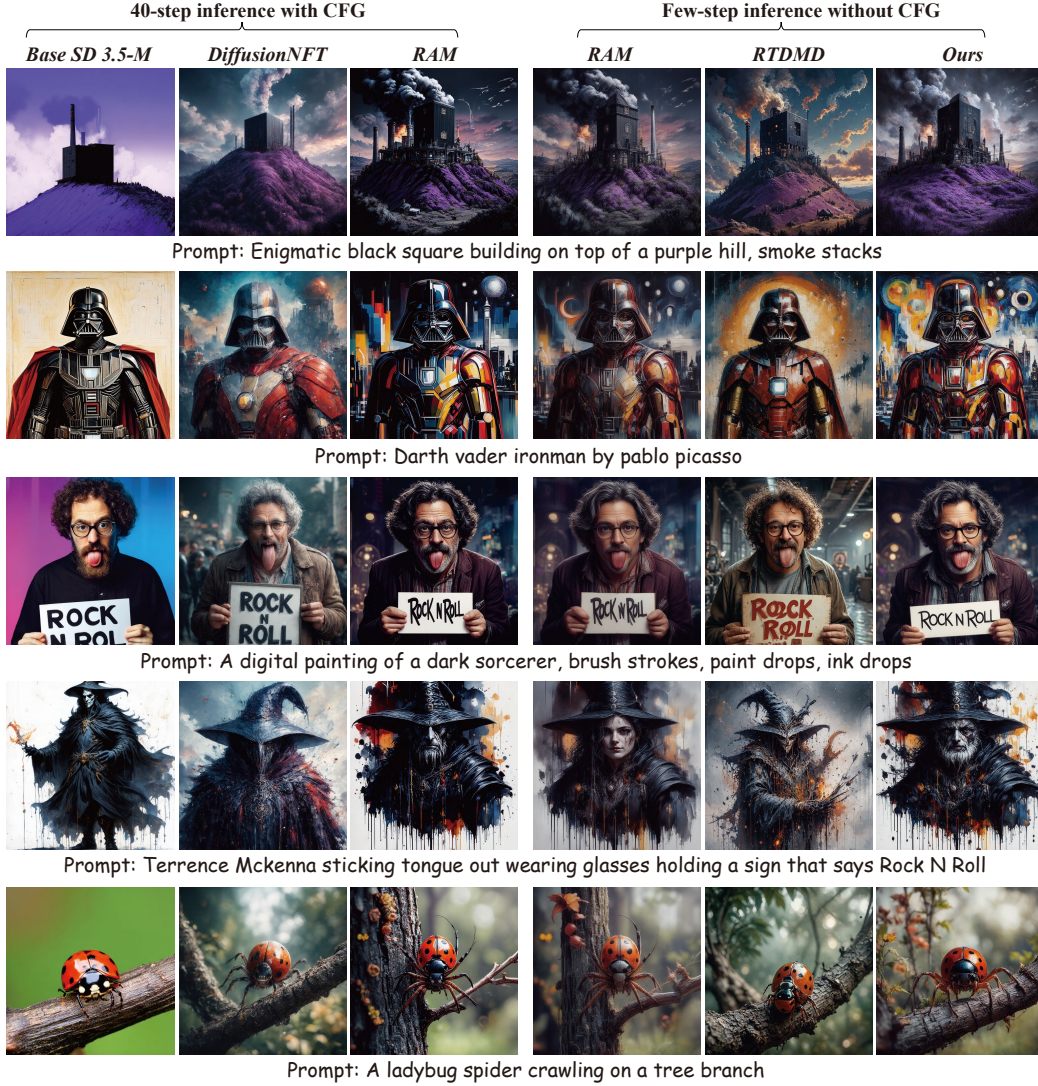


Figure 4: Qualitative comparison under PickScore alignment on the Pick-a-Pic testset.

Table 3: Progressive ablation of REST under PickScore training. All methods use 8-step CFG-free sampling on DrawBench dataset.

Variant	Teacher RL	Dual Branch	AMD	Aesthetic	DeQA	ImgRwd	HPSv2	PickScore
Base model	—	—	—	5.15	2.38	-0.96	0.18	20.39
RL	Yes	—	—	6.42	3.82	1.11	0.31	23.55
RL then Distill	Yes	—	—	6.22	4.14	1.32	0.32	23.83
RL & Distill	Yes	Yes	—	6.20	4.09	1.32	0.32	23.74
REST (full)	Yes	Yes	Yes	6.30	4.15	1.35	0.33	23.96

and thus potentially destabilizing early student training; a larger shift, in contrast, makes the objective closer to conservative imitation. As shown in Fig. 6, a moderate shift preserves the necessary imitation prior while turning clearly low-quality trajectories into repulsive supervision, thereby amplifying the effect of RL signals. We interpret this moderate negative modulation as a CFG-like contrastive force that is beneficial for improving generation quality. We also remove the EMA term

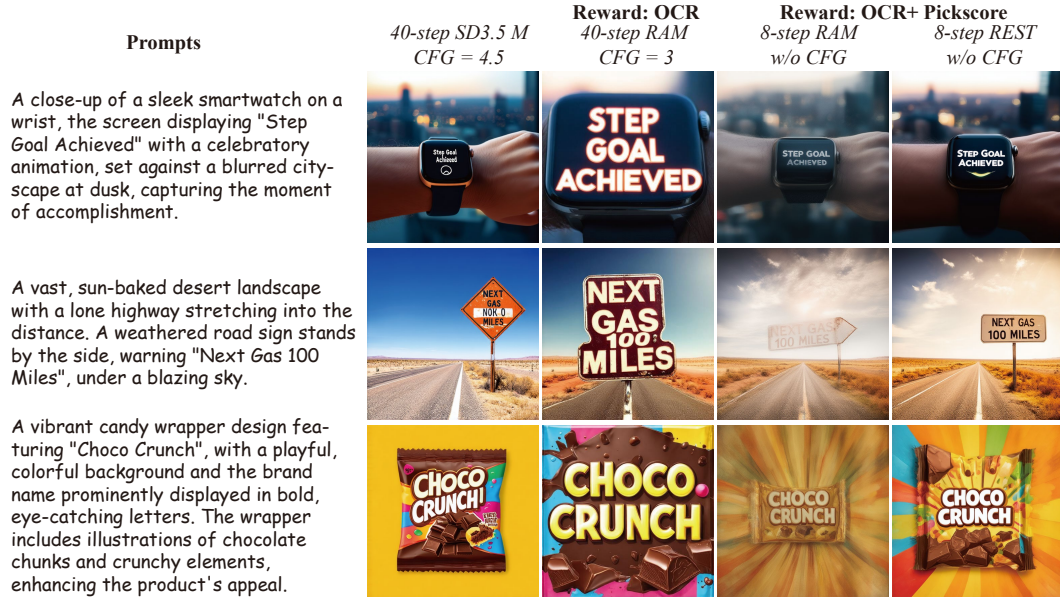


Figure 5: Qualitative comparison on visual text rendering. Single OCR reward leads to collapse by overly large text covering the whole image. OCR+PickScore optimization prevents the mode collapse, while REST achieves the highest image quality.



Figure 6: The shift b controls the positive imitation prior in the AMD weight. A moderate shift preserves the necessary imitation prior while turning clearly low-quality trajectories into repulsive supervision, thereby amplifying the effect/style from RL signals.

to evaluate its stabilizing role: without this anchor, the student exhibits much stronger performance oscillation rather than a stable improvement, confirming that EMA regularization helps absorb the jitter caused by the continuously evolving teacher policy.

Table 4: Ablation of the AMD shift b and EMA regularization under PickScore training, evaluated with DrawBench metrics.

Variant	Shift b	EMA	Aesthetic	DeQA	ImgRwd	HPSv2	PickScore
No shift	0	Yes	6.03	3.92	0.09	0.25	21.48
Larger shift	1.0	Yes	6.25	4.19	1.33	<u>0.32</u>	23.76
W/o EMA	0.5	No	6.32	4.11	<u>1.34</u>	0.31	<u>23.91</u>
REST (full)	0.5	Yes	<u>6.30</u>	<u>4.15</u>	1.35	0.33	23.96

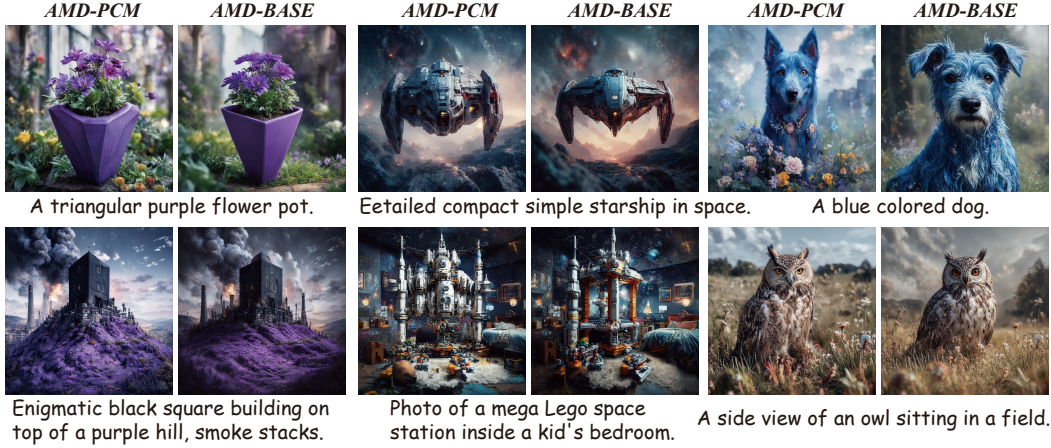


Figure 7: REST with different distillation objectives. AMD performs consistently with segment-velocity and PCM-style losses.

4.4 ANALYSIS

Training efficiency. Since REST reuses the teacher’s existing rollout trajectories and reward evaluations, the student branch incurs no additional sampling cost over the base RL pipeline—the most expensive stage in RL. Its only extra computation is the student forward/backward pass and EMA regularization. Specifically, in REST training, sampling rollout with reward computation accounts for **40.7%** of the wall-clock time, teacher training for **33.3%**, and student training for only **26.1%**. Taking into account overheads such as model checkpointing and online validation, the overall additional computational cost introduced by REST is **below 25%** over pure RL. Compared with sequential RL-then-distill training and RTDMD, both of which require substantially more training steps, REST also exhibits a clear advantage in convergence speed, as shown in Fig. 1.

Generality across distillation losses. AMD only modulates the weight of a base distillation objective and is therefore not tied to the per-step imitation loss (segment-velocity MSE). We further integrate AMD with Phased-Consistency-Model-style phase consistency. Specifically, within each student-defined phase, REST trains every adjacent pair of teacher states to predict the same phase endpoint (more expensive than per-step imitation loss): the online student predicts from the higher-noise state, while the EMA student provides a stop-gradient target from the lower-noise state, and the two predictions are matched with a pseudo-Huber loss. The reward-derived AMD weight is then applied to this phase-local consistency loss. As shown in Fig. 7, the PCM variant surpasses 8-step RAM in our experiments, and gets similar performance with segment-velocity MSE, implying REST’s compatibility with other distillation losses. Detailed procedures of the per-step imitation loss and PCM+AMD variants are provided in Algs. 1 and 2.

5 LIMITATIONS AND FUTURE WORK.

Although we validate AMD with both per-step imitation MSE and PCM-style consistency (Wang et al., 2024), its integration with DMD2 (Yin et al., 2024a) remains unexplored. A promising direction is to combine RL and DMD2 within a single REST stage to enable high-quality 4-step CFG-free inference. A second limitation is that our experiments instantiate the teacher branch only with RAM (Bergmeister et al., 2026). Nevertheless, REST explicitly decouples teacher-side RL from student-side distillation: the student consumes reward-scored ODE trajectories without depending on the teacher’s specific policy-optimization objective. Appendix A provides theoretical support for extending REST to DiffusionNFT (Zheng et al., 2025) by showing that representative ODE-based diffusion RL algorithms, including RAM and DiffusionNFT, share an anchored, reward-shifted velocity-regression structure despite differences in their anchor choices and hyperparameters. This structural commonality suggests that REST can serve as a general co-training framework for a broader class of ODE-based diffusion RL teachers. Empirically validating REST with DiffusionNFT and other policy-optimization algorithms remains important future work.

6 CONCLUSION

In this paper, we present **REST**, a unified RL-distillation co-training framework for efficient and reward-aligned diffusion generation. Our core insight is that reward-optimized diffusion rollouts already contain rich few-step distillation signals, but existing RL methods mainly optimize from terminal samples and leave the intermediate teacher trajectory underused. We operationalize this insight through a decoupled teacher–student design: the teacher branch continues standard RL post-training, while the student branch simultaneously learns a CFG-free few-step generator from the same reward-scored rollout trajectory. To make this distillation preference-aware, we introduce **Advantage-Modulated Distillation** (AMD), a general reward-aware wrapper that modulates arbitrary base distillation losses with advantage-derived weights, strengthening high-reward trajectory segments while suppressing or repelling low-reward ones. Experiments on SD3.5M across GenEval, OCR, and PickScore settings demonstrate that REST preserves the reward gains of a full-step RL teacher while substantially reducing inference cost, and the ablation and analysis studies further validate the importance of decoupled co-training, advantage modulation, KL-EMA stabilization, and the generality of REST across teacher RL algorithms and distillation losses.

REFERENCES

- Andreas Bergmeister, Stefanie Jegelka, Nikolas Nüsken, Carles Domingo-Enrich, and Jakiw Pidstrigach. Reinforce adjoint matching: Scaling rl post-training of diffusion and flow-matching models. *arXiv preprint arXiv:2605.10759*, 2026.
- Huayu Chen, Kaiwen Zheng, Qinsheng Zhang, Ganqu Cui, Yin Cui, Haotian Ye, Tsung-Yi Lin, Ming-Yu Liu, Jun Zhu, and Haoxiang Wang. Nft: Bridging supervised learning and reinforcement learning in math reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 124025–124042, 2026.
- Kevin Clark, Paul Vicol, Kevin Swersky, and David Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*, volume 2024, pp. 4793–4822, 2024.
- Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 2024.
- Linqian Fan, Peiqin Sun, Tiancheng Wen, Shun Lu, and Chengru Song. Rdm: Re-conceptualizing distribution matching as a reward for diffusion distillation. *arXiv preprint arXiv:2603.28460*, 2026.
- Black forest labs. Flux.1-dev. <https://github.com/black-forest-labs/flux>, 2024.

- Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, volume 36, 2023.
- Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 2022.
- Yushi Huang, Xiangxin Zhou, Ruoyu Wang, Chi Zhang, Jun Zhang, and Tianyu Pang. Reinforcing few-step generators via reward-tilted distribution matching. *arXiv preprint arXiv:2605.26108*, 2026.
- Dengyang Jiang, Dongyang Liu, Zanyi Wang, Qilong Wu, Liuzhuozheng Li, Hengzhuang Li, Xin Jin, David Liu, Zhen Li, Bo Zhang, et al. Distribution matching distillation meets reinforcement learning. *arXiv preprint arXiv:2511.13649*, 2025.
- Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, volume 36, 2023.
- Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.
- Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Yiming Cheng, Miles Yang, Zhao Zhong, and Liefeng Bo. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025.
- Zhanhao Liang, Yuhui Yuan, Shuyang Gu, Bohan Chen, Tiankai Hang, Mingxi Cheng, Ji Li, and Liang Zheng. Aesthetic post-training diffusion models from generic preferences with step-by-step preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13199–13208, 2025.
- Zhanhao Liang, Tao Yang, Jie Wu, Chengjian Feng, and Liang Zheng. Leapalign: Post-training flow matching models at any generation step by building two-step trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23238–23248, 2026.
- Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025.
- Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021.
- Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.
- Weijian Luo, Zemin Huang, Zhengyang Geng, J Zico Kolter, and Guo-jun Qi. One-step diffusion distillation through score implicit matching. *Advances in Neural Information Processing Systems*, 37:115377–115408, 2024.
- Thuan Hoang Nguyen and Anh Tran. Swiftbrush: One-step text-to-image diffusion model with variational score distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7807–7816, 2024.
- Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.
- Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, pp. 745–750, 2007.

- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.
- Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, volume 35, pp. 36479–36494, 2022.
- Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.
- Christoph Schuhmann and Romain Beaumont. Laion-aesthetics. laion.ai, 2022.
- Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv:2111.02114*, 2021.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 32211–32252, 2023.
- Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.
- Fu-Yun Wang, Zhaoyang Huang, Alexander W 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.
- Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025.
- 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.
- Xiaoshi Wu, Yiming Hao, Manyuan Zhang, Keqiang Sun, Zhaoyang Huang, Guanglu Song, Yu Liu, and Hongsheng Li. Deep reward supervisions for tuning text-to-image diffusion models. In *European Conference on Computer Vision*, pp. 108–124. Springer, 2024.
- Jiazheng 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. *NeurIPS*, 2023.
- Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning rl with pretraining in diffusion models. *arXiv preprint arXiv:2509.25050*, 2025a.
- Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrp: Unleashing grp on visual generation. *arXiv preprint arXiv:2505.07818*, 2025b.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T 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.

Zhiyuan You, Xin Cai, Jinjin Gu, Tianfan Xue, and Chao Dong. Teaching large language models to regress accurate image quality scores using score distribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14483–14494, 2025.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. *arXiv preprint arXiv:2509.16117*, 2025.

A RELATIONSHIP BETWEEN DIFFUSIONNFT AND RAM

Although our main experiments instantiate the teacher branch only with RAM, REST explicitly decouples teacher-side reinforcement learning from student-side distillation: the student consumes reward-scored ODE trajectories without depending on the teacher’s specific policy-optimization objective. This section provides theoretical support for extending REST to a broader class of ODE-based diffusion RL teachers by comparing RAM and DiffusionNFT. We show that these representative methods share an anchored, reward-shifted velocity-regression structure despite differences in their anchor choices and hyperparameters. Rather than claiming that the two algorithms are fully equivalent, this structural commonality identifies a shared trajectory interface that enables REST to accommodate different teacher optimizers without changing its student-side distillation mechanism.

To make this connection explicit, we compare their objectives on the forward noising process. Let $v^{\text{gt}} = \epsilon - x_0$ denote the flow-matching target, v^{old} the lagged policy used for sampling, and v_ϕ the trainable policy. DiffusionNFT constructs two implicit policies symmetric around v^{old} ,

$$v_\phi^+ = (1 - \beta)v^{\text{old}} + \beta v_\phi, \quad v_\phi^- = (1 + \beta)v^{\text{old}} - \beta v_\phi, \quad (16)$$

and optimizes the reward-weighted flow-matching objective

$$\mathcal{L}_{\text{NFT}} = r \|v_\phi^+ - v^{\text{gt}}\|^2 + (1 - r) \|v_\phi^- - v^{\text{gt}}\|^2, \quad (17)$$

where $r \in [0, 1]$ is obtained from the clipped group-relative advantage. For clarity, we first omit the time-dependent and detached self-normalization factors used in the implementation; their effect is discussed below.

Equivalent target of DiffusionNFT. Define

$$\Delta = v^{\text{old}} - v^{\text{gt}}, \quad \delta_\phi = v_\phi - v^{\text{old}}, \quad A = 2r - 1 \in [-1, 1]. \quad (18)$$

Equation 16 gives $v_\phi^+ - v^{\text{gt}} = \Delta + \beta\delta_\phi$ and $v_\phi^- - v^{\text{gt}} = \Delta - \beta\delta_\phi$. Substituting them into Eq. 17 yields

$$\mathcal{L}_{\text{NFT}} = r \|\Delta + \beta\delta_\phi\|^2 + (1 - r) \|\Delta - \beta\delta_\phi\|^2 \quad (19)$$

$$= \|\Delta\|^2 + 2\beta A \langle \Delta, \delta_\phi \rangle + \beta^2 \|\delta_\phi\|^2 \quad (20)$$

$$= \beta^2 \left\| \delta_\phi + \frac{A}{\beta} \Delta \right\|^2 + (1 - A^2) \|\Delta\|^2. \quad (21)$$

The last term is independent of v_ϕ . Therefore, Eq. 17 is gradient-equivalent to

$$\tilde{\mathcal{L}}_{\text{NFT}} = \beta^2 \|v_\phi - v_{\text{target}}^{\text{NFT}}\|^2, \quad v_{\text{target}}^{\text{NFT}} = v^{\text{old}} + \frac{A}{\beta} (v^{\text{gt}} - v^{\text{old}}). \quad (22)$$

Thus, DiffusionNFT can be interpreted as regression from the old policy toward a reward-dependent target. Positive advantages move the policy toward v^{gt} , while negative advantages extrapolate it away from v^{gt} through the same anchor v^{old} .

Comparison with RAM. RAM directly constructs the stop-gradient target

$$v_{\text{target}}^{\text{RAM}} = v^{\text{base}} + \eta A_{\text{RAM}} (v^{\text{gt}} - v^{\text{old}}), \quad \mathcal{L}_{\text{RAM}} = \|v_\phi - \text{sg}(v_{\text{target}}^{\text{RAM}})\|^2, \quad (23)$$

where v^{base} is the frozen pretrained velocity, A_{RAM} is the normalized advantage, and η is its scale. Equations 22 and 23 expose their shared template,

$$v_{\text{target}} = v_{\text{anchor}} + \gamma(A) (v^{\text{gt}} - v^{\text{old}}). \quad (24)$$

Both methods therefore perform forward-process velocity regression along the same reward-controlled direction $v^{\text{gt}} - v^{\text{old}}$. DiffusionNFT uses $v_{\text{anchor}} = v^{\text{old}}$ and $\gamma(A) = A/\beta$, whereas RAM uses $v_{\text{anchor}} = v^{\text{base}}$ and $\gamma(A) = \eta A_{\text{RAM}}$. In the practically relevant regime where the lagged policy remains close to the pretrained model, $v^{\text{old}} \approx v^{\text{base}}$, and the reward scales satisfy $A/\beta \approx \eta A_{\text{RAM}}$, their regression targets approximately coincide.

Algorithm 1 REST with Segment-Velocity Distillation and AMD

Require: Online/EMA teacher $v_\phi, v_{\bar{\phi}}$; online/EMA student $v_\theta, v_{\bar{\theta}}$; rewards $\{R_m\}_{m=1}^{N_R}$.
Require: Teacher steps M ; student boundaries $0 = q_0 < \dots < q_K = M$; AMD parameters $\lambda, b, \{\alpha_m\}$; EMA coefficient β_{ema} .
while not converged
 Phase 1: Teacher Rollout and Multi-Reward Advantage
 1: Sample prompt c and a full CFG-enabled trajectory with $v_{\bar{\phi}}$:
 $\mathcal{T} = \{x_{t_0}, \dots, x_{t_M}\}$, where $x_{t_{j+1}} = x_{t_j} + (t_{j+1} - t_j)v_{\bar{\phi}}^{\text{CFG}}(x_{t_j}, t_j, c)$.
 2: Decode x_{t_M} and compute $R_m(x_{t_M}, c)$ for every reward source m .
 3: Normalize and clip each reward within the same-prompt group to obtain $A^{(m)} \in [-1, 1]$.
 4: Fuse advantages: $A_{\text{mix}} = \sum_m \alpha_m A^{(m)} / \sum_m \alpha_m$.
 Phase 2: Decoupled Teacher RL Update
 5: Update ϕ with the original teacher RL objective using x_{t_M} and A_{mix} .
 6: Do not propagate student gradients into ϕ or the stored trajectory $\mathcal{T}$.
 Phase 3: Segment-Velocity Student Distillation
 7: **for** $k = 0, \dots, K - 1$ **do**
 8: Construct $s_k = (x_{t_{q_k}}, t_{q_k}, c)$ and segment target
 $a_k = (x_{t_{q_{k+1}}} - x_{t_{q_k}}) / (t_{q_{k+1}} - t_{q_k})$.
 9: Compute $w_{\text{AMD}} = \lambda(A_{\text{mix}} + b)$.
 10: Compute $\ell_{\text{seg}} = \|v_\theta(s_k) - \text{sg}(a_k)\|^2$.
 11: Compute $\ell_{\text{ema}} = \|v_\theta(s_k) - \text{sg}(v_{\bar{\theta}}(s_k))\|^2$.
 12: Accumulate $\mathcal{L}_{\text{student}}^{\text{seg}} += w_{\text{AMD}}\ell_{\text{seg}} + \beta_{\text{ema}}\ell_{\text{ema}}$.
 13: **end for**
 Phase 4: Independent Optimization and EMA Update
 14: Update only θ by minimizing $\mathcal{L}_{\text{student}}^{\text{seg}}$.
 15: $\bar{\phi} \leftarrow \rho_T \bar{\phi} + (1 - \rho_T)\phi$; $\bar{\theta} \leftarrow \rho_S \bar{\theta} + (1 - \rho_S)\theta$.
end while
Ensure: EMA student $v_{\bar{\theta}}$ for K -step CFG-free inference.

Full DiffusionNFT objective and limitations of the equivalence. The implementation of DiffusionNFT uses detached residual normalizers w_+ and w_- and an additional time weight. With the definitions above, its gradient has the form

$$\nabla_{v_\phi} \mathcal{L}_{\text{NFT}} \propto t \left[\left(\frac{r}{w_+} - \frac{1-r}{w_-} \right) \Delta + \beta \left(\frac{r}{w_+} + \frac{1-r}{w_-} \right) \delta_\phi \right]. \quad (25)$$

Because w_+ and w_- are stop-gradient quantities, this remains an anchored regression update: the first term supplies the reward-dependent direction and the second pulls the trainable policy toward v^{old} . When $w_+ = w_-$, Eq. 25 reduces exactly, up to a positive scalar, to the gradient of Eq. 22. When the two normalizers are merely close, this equivalence is approximate; when they differ substantially, they adaptively rescale the effective advantage and trust-region strength.

The relationship above establishes structural similarity rather than complete equivalence. DiffusionNFT parameterizes symmetric positive and negative policies, applies time-dependent self-normalization, and anchors its policy-improvement target at v^{old} . RAM instead uses a single closed-form stop-gradient target anchored at the frozen v^{base} , with a separately chosen advantage scale. Nevertheless, both optimize the same forward-process flow-matching direction through reward-shifted velocity regression. This shared structure explains why REST, although instantiated with RAM in our main experiments, can in principle use DiffusionNFT as its teacher optimizer without changing the student-side trajectory distillation mechanism.

B DETAILED TRAINING ALGORITHMS OF REST

We provide the complete training procedures for the two REST variants evaluated in this work. They share the same reward-optimized teacher branch, reuse the same teacher trajectories and terminal rewards, and differ only in the student-side base distillation objective wrapped by AMD. Let $\phi, \bar{\phi}, \theta$, and $\bar{\theta}$ denote the online teacher, EMA teacher, online student, and EMA student, respectively. Let M and K be the numbers of teacher and student steps, with $K \ll M$.

Algorithm 2 REST with PCM-Style Phase Consistency and AMD

Require: Online/EMA teacher $v_\phi, v_{\bar{\phi}}$; online/EMA student $v_\theta, v_{\bar{\theta}}$; rewards $\{R_m\}_{m=1}^{N_R}$.
Require: Teacher steps M ; phase boundaries $0 = q_0 < \dots < q_K = M$; AMD parameters $\lambda, b, \{\alpha_m\}$; EMA coefficient β_{ema} ; pseudo-Huber constant c_{huber} .
while not converged
 Phase 1: Teacher Rollout and Multi-Reward Advantage
 1: Sample prompt c and a full CFG-enabled trajectory $\mathcal{T} = \{x_{t_0}, \dots, x_{t_M}\}$ with $v_{\bar{\phi}}$.
 2: Decode x_{t_M} , compute all rewards, and obtain the fused advantage

$$A_{\text{mix}} = \sum_m \alpha_m A^{(m)} / \sum_m \alpha_m$$
 after per-reward normalization and clipping.
 Phase 2: Decoupled Teacher RL Update
 3: Update ϕ with its original RL objective; detach $\mathcal{T}$ from the teacher graph.
 Phase 3: PCM+AMD Student Distillation
 4: **for** $k = 0, \dots, K - 1$ **do**
 5: **for** $j = q_k, \dots, q_{k+1} - 1$ **do**
 6: Set the common phase endpoint to $t_{q_{k+1}}$.
 7: Predict the endpoint from the higher-noise state x_{t_j} with the online student:

$$\hat{x}_{k,j}^{\text{on}} = x_{t_j} + (t_{q_{k+1}} - t_j)v_\theta(x_{t_j}, t_j, c).$$

 8: Predict the same endpoint from the lower-noise state $x_{t_{j+1}}$ with the EMA student:

$$\hat{x}_{k,j}^{\text{ema}} = \text{sg}[x_{t_{j+1}} + (t_{q_{k+1}} - t_{j+1})v_{\bar{\theta}}(x_{t_{j+1}}, t_{j+1}, c)].$$

 9: Compute $\ell_{\text{pcm}}^{k,j} = \sqrt{\|\hat{x}_{k,j}^{\text{on}} - \hat{x}_{k,j}^{\text{ema}}\|_2^2 + c_{\text{huber}}^2} - c_{\text{huber}}$.
 10: Compute $w_{\text{AMD}} = \lambda(A_{\text{mix}} + b)$.
 11: Compute $\ell_{\text{ema}}^{k,j} = \|v_\theta(x_{t_j}, t_j, c) - \text{sg}(v_{\bar{\theta}}(x_{t_j}, t_j, c))\|^2$.
 12: Accumulate $\mathcal{L}_{\text{student}}^{\text{pcm}} += w_{\text{AMD}}\ell_{\text{pcm}}^{k,j} + \beta_{\text{ema}}\ell_{\text{ema}}^{k,j}$.
 13: **end for**
 14: **end for**
 Phase 4: Independent Optimization and EMA Update
 15: Update only θ by minimizing $\mathcal{L}_{\text{student}}^{\text{pcm}}$ over all phase-local pairs.
 16: $\bar{\phi} \leftarrow \rho_T \bar{\phi} + (1 - \rho_T)\phi$; $\bar{\theta} \leftarrow \rho_S \bar{\theta} + (1 - \rho_S)\theta$.
end while
Ensure: EMA student $v_{\bar{\theta}}$ for K -step CFG-free inference.

Preliminary: PCM-style phase consistency. The default REST variant above represents each student step by a single segment velocity between two selected teacher states. The PCM variant instead divides the fine-grained M -step teacher trajectory into K student phases and enforces consistency within each phase. The indices q_k and q_{k+1} denote the teacher-grid columns at the beginning and end of student phase k , respectively. Within this phase, $j \in \{q_k, \dots, q_{k+1} - 1\}$ indexes an adjacent pair of teacher states. Since the denoising schedule satisfies $t_j > t_{j+1}$, x_{t_j} is the higher-noise state and $x_{t_{j+1}}$ is the immediately following lower-noise state. All adjacent pairs in phase k share the same endpoint time $t_{q_{k+1}}$.

Starting from the higher-noise state, the online student predicts the phase-end latent as $\hat{x}_{k,j}^{\text{on}} = x_{t_j} + (t_{q_{k+1}} - t_j)v_\theta(x_{t_j}, t_j, c)$. Here, the superscript “on” denotes the online, trainable student, and the hat indicates that this is a predicted endpoint latent rather than an observed teacher state. From the lower-noise state, the EMA student analogously produces $\hat{x}_{k,j}^{\text{ema}}$, where “ema” denotes the lagged student and the prediction is treated as a stop-gradient target. In these expressions, $v_\theta(\cdot, t, c)$ is the conditional velocity predicted at time t for prompt c . PCM trains the two endpoint predictions to agree through the pseudo-Huber distance $\ell_{\text{pcm}}^{k,j}$. Intuitively, regardless of which nearby state within a phase is used as the starting point, the student should predict the same phase endpoint. REST then applies the rollout-level AMD weight $w_{\text{AMD}} = \lambda(A_{\text{mix}} + b)$ to this consistency loss, strengthening phase-consistency learning on preferred trajectories and weakening or reversing it on low-reward trajectories.

Shared rollout and decoupled optimization. Both variants incur no additional image sampling or reward-model evaluation beyond the teacher RL pipeline. The trajectory and terminal rewards are collected once by the teacher and reused by both optimizers. Although the teacher is updated before the student in each training iteration, the student loss is evaluated on the stored pre-update trajectory and never backpropagates into the teacher. REST therefore preserves the teacher optimizer exactly and changes only how the student consumes its reward-scored rollout.

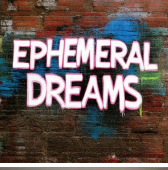
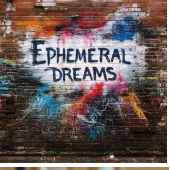
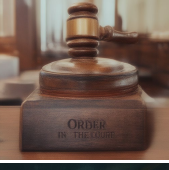
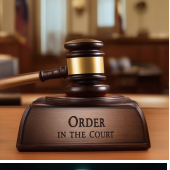
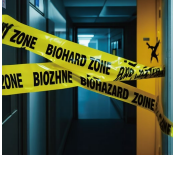
Prompts	40-step SD3.5 M CFG = 4.5	Reward: OCR 40-step RAM CFG = 3	Reward: OCR+ Pickscore 8-step RAM w/o CFG	8-step REST w/o CFG
A rustic scarecrow stands in a golden wheat field, its worn shirt painted with bold black letters: "No Crows Allowed". The sun sets behind, casting long shadows and a warm glow over the scene, emphasizing the scarecrow's vigilant stance.				
A realistic photograph of a brick wall with graffiti reading "Ephemeral Dreams" partially obscured by layers of faded, multi-colored spray paint, capturing the transient nature of street art.				
A courtroom scene with a judge's gavel resting on a wooden plaque that reads "Order in the Court", set against the backdrop of a quiet, solemn courtroom.				
A realistic photograph of a scene with bright yellow caution tape, prominently displaying the text "Biohazard Zone", stretched across a doorway in a dimly lit laboratory.				

Figure 8: Additional qualitative results on visual text rendering.

Difference between the two student objectives. The default variant uses one finite-difference velocity target for each student phase. It directly teaches the student to traverse the complete phase in one step and requires K student targets per trajectory. PCM+AMD instead enumerates every adjacent teacher pair inside each phase. The online and EMA students start from different noise levels but are constrained to predict the same phase endpoint. If the selected student boundaries cover the complete M -step teacher trajectory, the PCM variant uses $\sum_k (q_{k+1} - q_k) = M$ consistency pairs rather than K segment targets. This explains its higher forward/backward cost.

Role of AMD and EMA. In both variants, AMD multiplies the per-sample base distillation loss; it does not modify the segment velocity or PCM endpoint target. Consequently, positive $A_{\text{mix}} + b$ strengthens imitation or consistency on preferred trajectories, whereas a negative coefficient reverses the corresponding gradient and discourages low-reward trajectories. The shift b supplies a positive imitation prior during early training. The EMA student plays two related but distinct roles: in the default variant it is an explicit stability regularizer, while in PCM+AMD it additionally provides the stop-gradient phase-consistency target. The online student is the only student branch updated by backpropagation in either case.

Additional qualitative results for visual text rendering and compositional generation are shown in Figs. 8 and 9, respectively.



Figure 9: Additional qualitative results on compositional generation (GenEval task).