OMNI-DIFFUSION-DISTILL: FEW-STEP DISTILLATION OF UNIFIED MULTIMODAL DIFFUSION LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

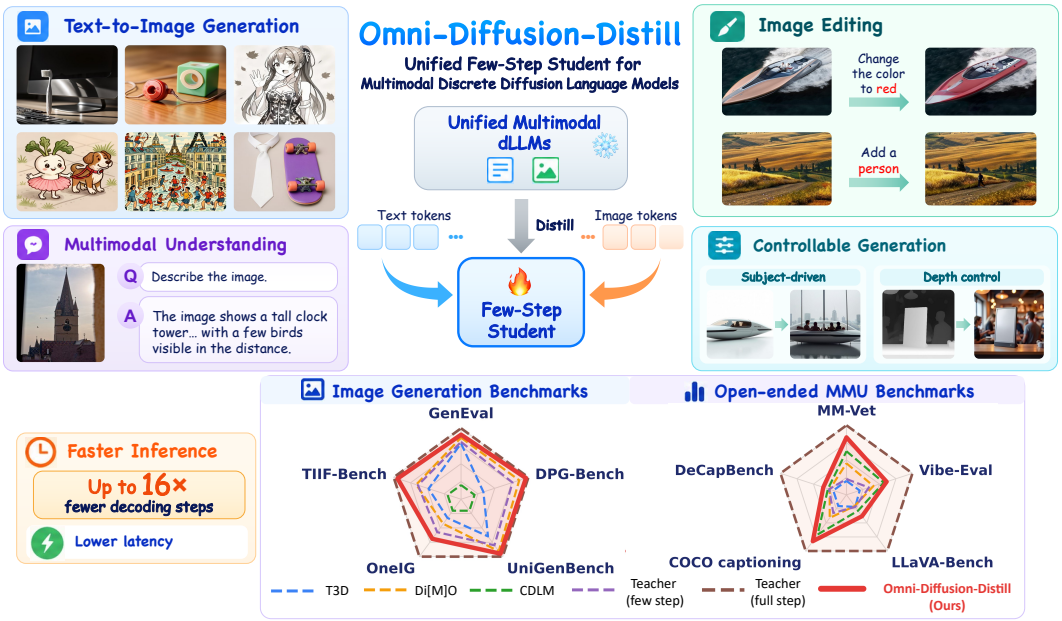


Figure 1: **Few-step distillation of a unified multimodal dLLM.** With up to 16× fewer decoding steps, our student dLLM unifies four tasks: text-to-image generation, image editing, controllable generation and multimodal understanding.

ABSTRACT

Unified multimodal diffusion large language models (dLLMs) offer a single architecture for both image generation and multimodal understanding, but their iterative decoding requires tens to hundreds of forward passes. Existing few-step distillation methods largely focus on either image generation or text generation, making it unclear how to compress a fully discrete multimodal dLLM into a single efficient student while preserving both generation and understanding. We introduce **Omni-Diffusion-Distill**, a unified two-stage distillation framework that retains strong generation and understanding capabilities while substantially reducing the inference cost of a unified multimodal dLLM. Omni-Diffusion-Distill aligns the distillation of both generation and understanding, for both images and text, in the discrete token space. In the first stage, the student is trained to skip decoding steps by replaying cached teacher trajectories, and in the second stage the student is refined on intermediate states along its own rollouts. We further remedy two sources of degradation in unified distillation with a pairwise collision penalty that reduces repetition under parallel text decoding, and entropy-matched guidance that prevents entropy collapse caused by fitting the sharpened teacher distribution in image generation. Omni-Diffusion-Distill achieves state-of-the-art trade-offs between decoding efficiency and generation and understanding performance for multimodal dLLMs, reducing image generation from 128 to 8 decoding steps and multimodal understanding from 512 to 64, giving **18.2**× and **21.2**× wall-clock speedups. Under these budgets, it scores **0.828** on GenEval and **83.0** on DPG-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Bench for text-to-image generation, while reaching GPT judge scores of **20.0** on MM-Vet and **57.2** on COCO captioning (twice the teacher's 28.4 at the same steps) for multimodal understanding.

1 INTRODUCTION

Building a single model that can both understand and generate across text and images has become an important direction in multimodal learning. *Diffusion large language models (dLLMs)* provide a natural foundation for this goal: they generate by iteratively decoding masked tokens with a bidirectional model (Austin et al., 2021; Sahoo et al., 2024; Nie et al., 2025), while vector-quantized image tokenizers allow images to be represented in the same discrete token space as text (van den Oord et al., 2017; Esser et al., 2021). Recent *unified multimodal dLLMs* exploit this formulation to support both image generation and multimodal understanding within a single network (Yang et al., 2026; Xin et al., 2025). Their flexibility, however, comes with a substantial inference cost: producing one output can require tens to hundreds of sequential decoding steps. *Distillation* is the standard way to shorten an iterative sampler, and for a unified model it has to shorten the decoding of images and of text at once.

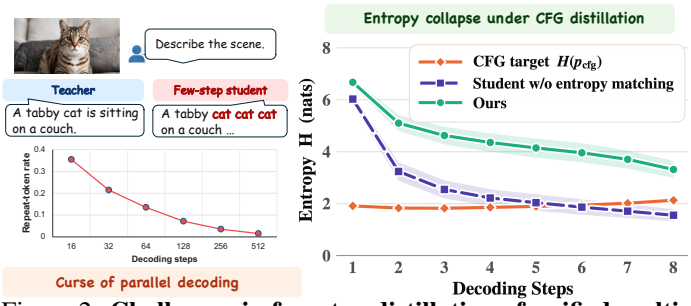


Figure 2: **Challenges in few-step distillation of unified multimodal dLLMs. Left:** Decoding more text tokens in parallel ignores their dependencies and increases repetition. **Right:** A student distilled toward the low-entropy CFG target (orange) keeps losing entropy and ends below it (purple). Entropy-matched guidance keeps our student's entropy high (green).

However, existing few-step distillation methods largely treat image generation (Yin et al., 2024b; Song et al., 2023; Zhu et al., 2025a) and text dLLMs (Kim et al., 2025; Deschenaux & Gulcehre, 2025; Zhang et al., 2026) separately, and directly applying them to a unified model raises two modality-specific challenges, illustrated in Figure 2. *First*, for text, reducing the number of decoding steps requires committing more tokens in parallel. Tokens committed in the same decoding step are sampled from independent per-position predictions and therefore cannot condition on one another, leading to the *curse of parallel decoding* (Wu et al., 2025d; Liu et al., 2025a; Kang et al., 2026). This problem becomes increasingly severe under aggressive step reduction and often manifests as repetition in long-form generation. *Second*, for image generation, removing the unconditional forward pass at inference requires the student to internalize classifier-free guidance (CFG) (Ho & Salimans, 2022) during distillation. CFG sharpens the teacher distribution throughout decoding (Sanchez et al., 2023), and repeatedly matching these sharpened targets progressively lowers the entropy of the student's own predictions. This loss of entropy reduces sample diversity (Ho & Salimans, 2022; Yin et al., 2024b). In image editing, it makes the student tend to leave the source image unchanged (Brooks et al., 2023). Therefore, few-step distillation of a unified dLLM requires not only a common training framework across modalities, but also mechanisms tailored to the distinct challenges introduced by their decoding processes. This raises a core question:

Can we unify the distillation of image and text generation in a multimodal dLLM?

We answer this question with **Omni-Diffusion-Distill**, a unified few-step distillation framework for fully discrete multimodal dLLMs. In these models, a single denoiser predicts a categorical distribution over tokens for both text and images. Image and text decoding can therefore share one distillation framework. We train Omni-Diffusion-Distill in two stages. *First*, an off-policy stage replays cached teacher trajectories from which the student learns to skip decoding steps. *Second*, an on-policy stage refines the student on intermediate states along its own rollouts. Importantly, we address the degradation of unified distillation with two remedies: a *pairwise collision penalty* that penalizes the probability that two text positions decoded in parallel commit the same token, and *entropy-matched guidance* that rescales the CFG target so that its entropy matches that of the

conditional prediction. Figure 1 shows the overall achievements of Omni-Diffusion-Distill, and Figure 3 gives an overview of the framework. We summarize our contributions as follows:

- **Image generation task.** At 8 decoding steps, Omni-Diffusion-Distill reaches overall scores of 0.828 on GenEval and 83.0 on DPG-Bench, ahead of every distilled baseline on all five text-to-image benchmarks and within 0.8 points of the full-step teacher on DPG-Bench. On Graph-200K it reaches a DINOv2 image similarity of 0.789 against the full-step teacher's 0.799.
- **Multimodal understanding task.** At 64 decoding steps, Omni-Diffusion-Distill reaches GPT judge scores of 20.0 on MM-Vet and 57.2 on COCO captioning, twice the same-step teacher's 28.4. It leads every distilled baseline on all five open-ended benchmarks.
- **Efficiency.** The student generates an image in 8 steps and answers a question in 64. On one A100 this is 18.2× faster than the teacher for text-to-image generation and 21.2× for understanding.

2 Related Work

Unified models differ in how images are represented: a shared discrete vocabulary (Team, 2024; Wang et al., 2024), a separate tokenizer for generation (Wu et al., 2025b), or continuous latents generated with diffusion (Zhou et al., 2025; Deng et al., 2025). Closest to us are those that generate with an iterative discrete denoiser over a shared text and image vocabulary (Yang et al., 2026; Xin et al., 2025; Swerdlow et al., 2025; Shi et al., 2025). Few-step distillation methods are largely modality-specific: trajectory-based distillation and distribution matching for image diffusion (Salimans & Ho, 2022; Song et al., 2023; Liu et al., 2023b; Yin et al., 2024b;a; Zhou et al., 2024), output re-masking for discrete image generators (Zhu et al., 2025a), objectives that train on student-generated sequences (Agarwal et al., 2024; Ko et al., 2024) and preference objectives (Rafailov et al., 2023) for autoregressive LLMs, and, for text dLLMs, trajectory-based distillation (Kim et al., 2025; Deschenaux & Gulcehre, 2025; Zhang et al., 2026) and distribution matching (Zheng et al., 2026). Unified models have been accelerated by consistency distillation (Xu et al., 2026) and by speculative decoding with a distilled continuous image branch (Lu et al., 2025). Omni-Diffusion-Distill instead performs on-policy refinement on states the student itself produces, and treats both modalities as masked tokens of one model, so one framework serves both. Appendix A expands this discussion.

3 Method

3.1 Background and Preliminaries

Why a fully discrete unified model. Omni-Diffusion-Distill leverages a single denoiser that predicts a categorical distribution at every position, for both text and image tokens. At every step, the teacher therefore gives an explicit prediction, and one distillation framework can learn from it directly. Most unified models do not have such a denoiser. Some generate images with a continuous diffusion branch or decoder (You et al., 2026b; Li et al., 2025d; AI et al., 2026). Some use separate parameters for the two modalities (Li et al., 2026d). Others replace masked diffusion with discrete flow matching (Wang et al., 2026a; Luo et al., 2026). Only a few unified models are fully discrete (Yang et al., 2026; Shi et al., 2025; Swerdlow et al., 2025; Xin et al., 2025). Our method is not designed around a specific architecture. It only requires a shared discrete token space, iterative masked decoding and classifier-free guidance for images.

Notation. Text tokens, vector-quantized image tokens and a mask token share one vocabulary. Masked diffusion learns to recover masked output tokens. Decoding starts from a fully masked output. At each step, the model commits some masked positions, replacing their mask tokens with sampled tokens that stay fixed. We call the partially decoded output at decoding step t the state x_t, and write $\mathcal{M}(x_t)$ for its masked positions. At each masked position i, the frozen teacher p_ϕ and the student q_θ predict $p_\phi^i(\cdot \mid x_t)$ and $q_\theta^i(\cdot \mid x_t)$.

Block-causal attention. The teacher attends bidirectionally over the whole sequence. The student uses a block-causal mask for text outputs and bidirectional attention for images. Every position can attend to the prompt. Text outputs are split into blocks, and each text token attends to its own block and to all earlier blocks. At inference, the student decodes all image tokens in parallel and commits the most confident positions at each step. Text is decoded one block at a time, so the KV cache of finished blocks can be reused exactly.

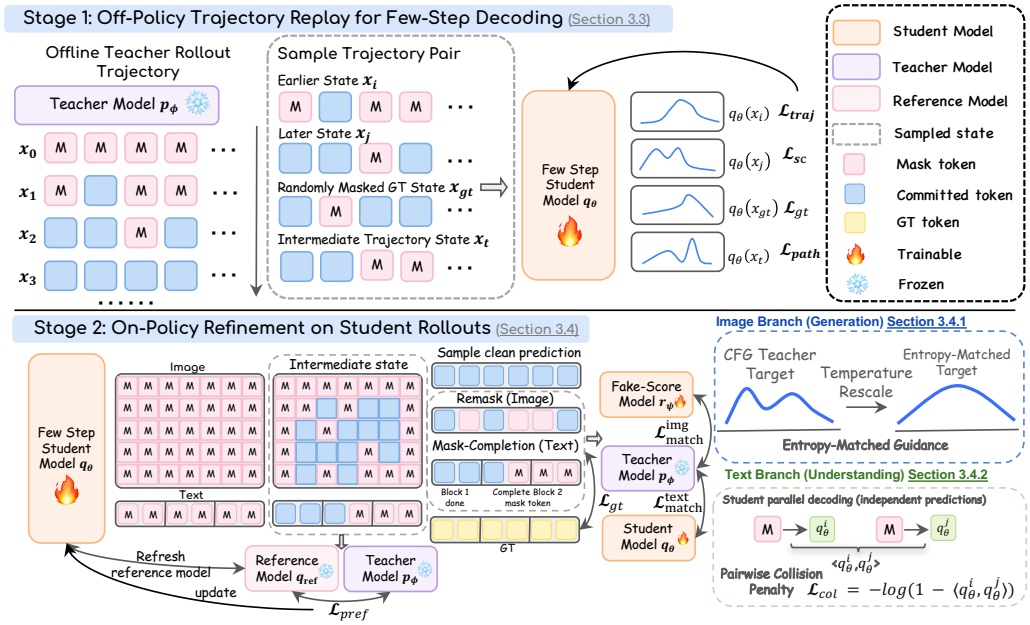


Figure 3: **Overview of Omni-Diffusion-Distill.** Stage 1 replays cached teacher trajectories off-policy, and Stage 2 refines the student on its own rollouts. Entropy-matched guidance rescales the image teacher's entropy, and a pairwise collision penalty reduces text repetition.

3.2 Motivation for Two-Stage Training

Stage 1 teaches the student to decode in fewer steps. The student is initialized from the teacher, but it was never trained to decode in so few steps. With fewer steps, it must commit more tokens in each forward pass. Stage 1 teaches this offline from teacher trajectories cached in advance. After Stage 1, the student can decode in fewer steps, but its quality is still below the teacher's (Table 19).

Stage 2 reduces the remaining quality gap. Stage 1 trains only on states from the teacher's trajectories. At inference, however, the student continues from states produced by its own earlier steps, so its errors can accumulate. Stage 2 addresses this by training the student on intermediate states of its own rollouts. This stage also addresses the two challenges of repeated text and entropy collapse.

3.3 Stage 1: Off-Policy Trajectory Replay for Few-Step Decoding

Cached teacher trajectory. For each training example, we run the teacher once offline and cache its trajectory. The cache records the step at which each token is committed and a few keyframes along the trajectory. Each keyframe stores the state at that step and the teacher's predicted distributions at the positions committed since the previous keyframe. On image tasks, we merge guidance into the teacher logits before storing them, $\ell^{\text{cfg}} = \ell^{\text{uncond}} + w_{\text{task}}\,(\ell^{\text{cond}} - \ell^{\text{uncond}})$. Here ℓ^{cond} and ℓ^{uncond} are the teacher's logits at a masked position with and without the prompt, and w_{task} is the guidance strength of the task. The student therefore learns the guided distribution directly.

Objective. Stage 1 optimizes the same four terms for every task,

$$\mathcal{L}_1 = \lambda_{\text{traj}}\,\mathcal{L}_{\text{traj}} + \lambda_{\text{sc}}\,\mathcal{L}_{\text{sc}} + \lambda_{\text{path}}\,\mathcal{L}_{\text{path}} + \lambda_{\text{gt}}\,\mathcal{L}_{\text{gt}}, \tag{1}$$

where the λ are scalar weights. Each training step samples two keyframes of a cached trajectory. The trajectory term $\mathcal{L}_{\text{traj}}$ is the forward KL divergence between the cached teacher predictions and the student's predictions on the earlier keyframe, at the positions the teacher commits between the two keyframes. Since the teacher made these predictions at later steps, the student learns to commit in one step what the teacher commits over several. The self-consistency term $\mathcal{L}_{\text{sc}}$ uses the student's prediction at the later keyframe as a fixed target for its prediction at the earlier one, at the positions still masked at the later keyframe. This keeps the student consistent when it skips ahead along the trajectory. The path term $\mathcal{L}_{\text{path}}$ rebuilds an intermediate state at a random step from the cached commit step of each token and trains the student to predict the tokens the teacher finally commits at its masked positions. The ground-truth anchor $\mathcal{L}_{\text{gt}}$ is the masked cross-entropy on the training data. Eqs. 5 and 6 in Appendix B give all four terms in full, and Algorithm 3 gives the training loop.

3.4 Stage 2: On-Policy Refinement on Student Rollouts

Stage 2 starts from the Stage-1 student (Algorithm 1). Each training step samples one task and rolls out the student without gradient (line 4). We build a state s from the rollout, run the teacher on s, and train the student to match the teacher prediction at the scored positions $\mathcal{M}$. Both modalities share this loop and differ in the state, the teacher prediction and the matching term, so $\mathcal{L}_{\text{match}} = \mathbb{E}_{\text{task}}[\mathcal{L}^{\text{task}}_{\text{match}}]$.

Algorithm 1 Stage 2: On-Policy Refinement

1: **Input:** teacher p_ϕ, student q_θ, auxiliary network $r_\psi \leftarrow p_\phi$, reference $q_{\text{ref}} \leftarrow q_\theta$, weights λ, training steps N_2, refresh interval τ, stop-gradient sg
2: **Output:** few-step student q_θ
3: **for** N_2 training steps **do**
4: sample a task; roll out q_θ ▷ no grad
5: **if** image task **then**
6: $z_\theta \leftarrow$ logits of q_θ at a random rollout state x_t ▷ grad
7: $\hat{x}_0 \sim \text{softmax}(\text{sg}[z_\theta])$; $s \leftarrow \text{remask}(\hat{x}_0)$
8: $\mathcal{M} \leftarrow$ positions masked in both s and x_t
9: $p_\phi \leftarrow$ entropy-matched teacher on s ▷ Eq. 2
10: $\nabla_{z_\theta} \mathcal{L}^{\text{img}}_{\text{match}} \leftarrow \text{sg}[r_\psi - p_\phi]$
11: $\mathcal{L}_{\text{col}} \leftarrow 0$
12: **else**
13: $s \leftarrow$ rollout state at a random decoding step of a block
14: $\mathcal{M} \leftarrow$ uncommitted positions of s
15: $p^{\text{seq}}_\phi \leftarrow p_\phi$ decodes $\mathcal{M}$ one at a time
16: $\mathcal{L}^{\text{text}}_{\text{match}} \leftarrow \text{JSD}(p^{\text{seq}}_\phi, q_\theta)$ ▷ grad
17: $\mathcal{L}_{\text{col}} \leftarrow$ collision penalty on $\mathcal{P}$ ▷ Eq. 3
18: **end if**
19: $\mathcal{L}_{\text{pref}} \leftarrow$ preference term ▷ Eq. 8
20: $\mathcal{L}_{\text{gt}} \leftarrow$ CE on freshly masked data
21: $\theta \leftarrow \text{update}(\theta, \mathcal{L}_2)$ ▷ λ-weighted sum
22: $\psi \leftarrow \text{update}(\psi, \text{CE}(r_\psi, \hat{x}_0))$ on image tasks
23: every τ steps: $q_{\text{ref}} \leftarrow q_\theta$
24: **end for**

3.4.1 Image Branch: Matching the Distribution

This branch trains the student to match the teacher's image distribution (lines 6–11). The student first predicts on a rollout state x_t (line 6). We sample a clean image $\hat{x}_0$ from this prediction and re-mask it to obtain the scored state s (line 7). The visible tokens of s are thus generated by the student. The student's prediction on s does not capture the distribution of images it generates, so following distribution matching for diffusion models (Yin et al., 2024b;a; Zhu et al., 2025a), we estimate this distribution with an auxiliary network r_ψ. The student is updated only at the scored positions $\mathcal{M}$, which are masked in both s and x_t (line 8). Sampling discrete tokens is not differentiable, so we cannot backpropagate through $\hat{x}_0$. Instead, we directly set the gradient on the student's logits z_θ to the difference between the predictions of r_ψ and the teacher on s, at each position in $\mathcal{M}$ (line 10). Meanwhile, r_ψ is trained with cross-entropy on the student's samples, $\text{CE}(r_\psi(\cdot \mid s), \hat{x}_0)$, so that it keeps up with the student.

Entropy-matched guidance. The teacher prediction here is the CFG target $\text{softmax}(\ell^{\text{cfg}})$. It is much sharper than the conditional prediction $\text{softmax}(\ell^{\text{cond}})$ at every step, and its entropy changes little along the trajectory (Figure 10). A student trained to fit it keeps losing entropy over the decoding steps and ends even sharper than the target (Figure 2 and Table 18). Such a sharp student tends to leave the source image unchanged in editing. We therefore rescale the CFG logits with a temperature ν (line 9). At each scored position, we choose ν so that the entropy H of the rescaled target is a fixed fraction ρ of the entropy of the conditional prediction,

$$H\big(\text{softmax}(\ell^{\text{cfg}}/\nu)\big) = \rho\, H\big(\text{softmax}(\ell^{\text{cond}})\big), \qquad p_\phi \leftarrow \text{softmax}\big(\ell^{\text{cfg}}/\nu\big). \tag{2}$$

We solve for ν by bisection (Appendix B). The rescaled distribution p_ϕ is then used as the teacher prediction in line 10. Temperature scaling changes only the entropy and keeps the ranking of tokens.

3.4.2 Text Branch: Sequential Targets and Penalty on Parallel Collisions

The text branch differs from the image branch in two ways (lines 13–17). First, it matches the teacher at each position instead of over the whole output. The student predicts these per-position distributions directly, so no auxiliary network is needed. Second, its main challenge is repetition under parallel decoding. We take s as the rollout state at a random step (line 13), and $\mathcal{M}$ as the positions not yet committed in s (line 14).

Sequentially conditioned teacher predictions. Running the teacher once on s gives its one-step marginal $p^i_\phi(\cdot \mid s)$ at each position. This marginal does not know which tokens the other positions will commit. We instead let the teacher decode the scored positions one at a time (line 15). At each position i, the teacher commits a token, and we record the distribution it used, denoted $p^{\text{seq},i}_\phi$. This distribution is conditioned on the tokens the teacher has already committed. The student therefore

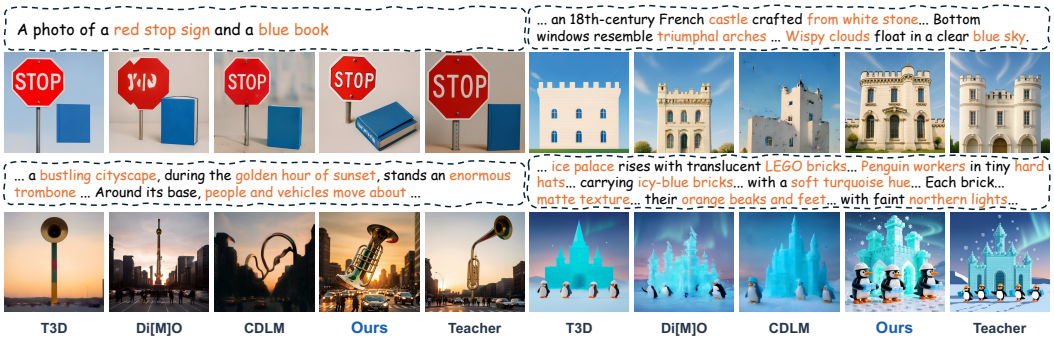


Figure 4: **Qualitative comparison on T2I.** All methods use the same prompts and seeds. Distilled models use 8 decoding steps, and the teacher uses 128.

learns from sequential context, while its own decoding stays parallel. The matching term is the Jensen-Shannon divergence between $p_\phi^{\text{seq},i}$ and q_θ^i, averaged over the positions in $\mathcal{M}$ (line 16).

Penalty on repetitions in parallel decoding. $\mathcal{L}_{\text{match}}^{\text{text}}$ is averaged over positions, so it is already minimal when each position matches its own teacher prediction. However, positions committed in the same decoding step are sampled independently, and the matching term does not penalize repeated tokens among them. When a token is plausible at two positions of a block, the student may commit it at both. We call this chance the collision probability. For positions i and j, it equals the inner product $\langle q_\theta^i, q_\theta^j \rangle$. All scored positions of a block are predicted from the same state s, and any two of them may be committed in the same decoding step. We therefore penalize collisions over the set $\mathcal{P}$ of all pairs of scored positions (line 17),

$$\mathcal{L}_{\text{col}} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} -\log\left(1 - \langle q_\theta^i, q_\theta^j \rangle\right). \tag{3}$$

The penalty reduces but does not forbid repetition. A token can still appear at two positions if the student assigns it a high probability at both. We apply it only to text, where parallel decoding often causes repetition.

Stage-2 objective. Stage 2 optimizes $\mathcal{L}_2 = \lambda_{\text{match}}\mathcal{L}_{\text{match}} + \lambda_{\text{col}}\mathcal{L}_{\text{col}} + \lambda_{\text{pref}}\mathcal{L}_{\text{pref}} + \lambda_{\text{gt}}\mathcal{L}_{\text{gt}}$. Here $\mathcal{L}_{\text{match}}$ is the expectation over tasks defined above, $\mathcal{L}_{\text{col}}$ is Eq. 3, and $\mathcal{L}_{\text{gt}}$ is the same ground-truth anchor as in Stage 1. The preference term $\mathcal{L}_{\text{pref}}$ follows T3D (Zhang et al., 2026) (Appendix B).

4 Experiments

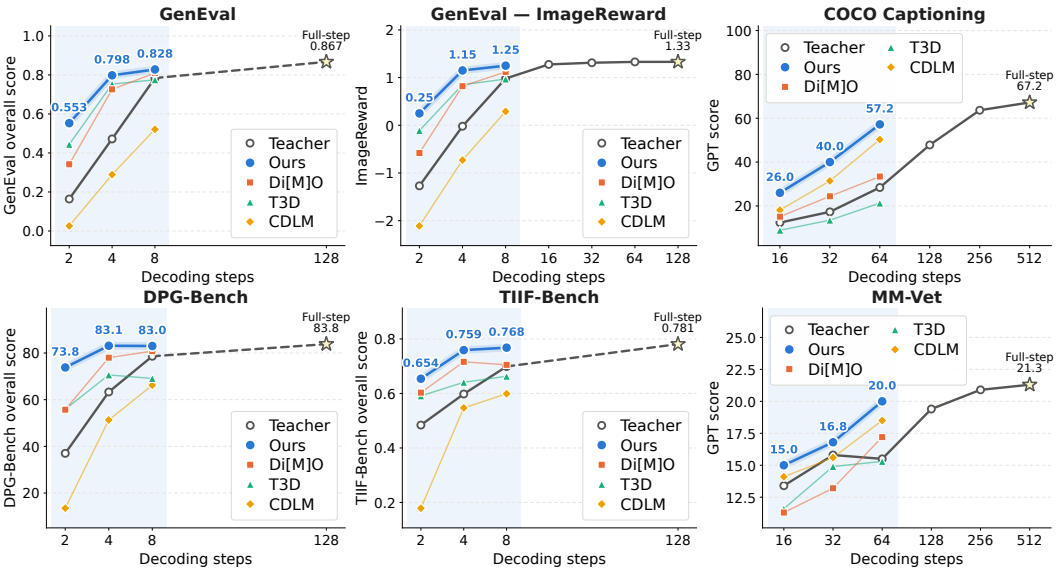


Figure 5: **Performance across decoding budgets.** The first two columns show T2I benchmarks, and the last column shows MMU benchmarks. The teacher curve extends to the full-step budget.

Table 1: **Quantitative Results on T2I Benchmarks.** Overall score on each benchmark. ImageReward (Xu et al., 2023) and Aesthetic are image quality metrics computed on GenEval. **Best** and second mark the top two distilled results.

Method	NFE ↓	GenEval ↑			DPG-Bench	UniGenBench	OneIG	TIIF-Bench
		Overall	ImageReward	Aesthetic				
Teacher (full-step)	128×2	0.867	1.33	3.65	83.8	0.722	0.445	0.781
T3D	8×1	0.775	0.97	3.55	69.1	0.640	0.306	0.663
Di[M]O	8×1	0.810	1.12	3.61	80.8	0.700	0.327	0.705
CDLM	8×1	0.521	0.29	3.33	66.2	0.536	0.280	0.599
Omni-Diffusion-Distill	8×1	**0.828**	**1.25**	**3.71**	**83.0**	**0.710**	**0.381**	**0.768**
Teacher (few-step)	8×2	0.784	0.98	3.61	78.6	0.676	0.357	0.699

4.1 Experimental Setup

Model and data. We distill Lumina-DiMOO (Xin et al., 2025), an 8B fully discrete unified dLLM whose single network covers all four tasks. Training uses approximately 51K examples, and Appendices C.1 and C.2 give the data sources and training settings.

Benchmarks and baselines. Following the teacher's protocol for most benchmarks, we evaluate the four tasks in three groups on 17 benchmarks: text-to-image generation (T2I), image-to-image generation (I2I) and multimodal understanding (MMU). I2I covers image editing and controllable generation. MMU uses both discriminative and open-ended benchmarks. We also report the teacher at both the few-step and full-step budgets. The distilled baselines are Di[M]O (Zhu et al., 2025a), CDLM (Kim et al., 2025) and T3D (Zhang et al., 2026). Appendices C.3 and C.4 give more details.

Decoding budgets and cost. Image tasks use 8, 4 or 2 decoding steps, against 128 for the teacher. Understanding uses 64, 32 or 16 steps, against 512 for the teacher. We report the 8-step and 64-step results here and the smaller budgets in Appendix D.1. Figure 5 shows how performance changes with the decoding steps on T2I and MMU benchmarks. We also report the number of function evaluations (NFE), the decoding steps times the forward passes per step. The student uses one pass per step, while the teacher adds one pass per guidance term, giving two on T2I and three on I2I.

4.2 T2I: Outperforming Every Distilled Baseline and the Few-Step Teacher

Table 1 reports the overall scores on five T2I benchmarks. They cover compositional prompts, dense prompt following, text rendering, reasoning, style and long instructions. Figure 4 compares samples under the same prompts and seeds. Omni-Diffusion-Distill is the best distilled method on all five benchmarks. At 8 steps, it reaches 0.828 on GenEval and 83.0 on DPG-Bench. It is above the few-step teacher on every benchmark at half its NFE, and within 0.8 DPG-Bench points of the full-step teacher. With fewer steps, its lead over the baselines generally becomes larger. Appendix D.2 gives more results.

4.3 I2I: Editing Degrades Little Under Step Reduction

Table 2 reports image editing on the I2I benchmark and controllable generation on Graph-200K. Figure 6 shows four edit types, and Figure 9 shows more results. **Editing quality drops little with fewer steps.** When the teacher goes from 128 to 8 steps, the I2I benchmark drops by 2.6% and Graph-200K by 0.9%, consistent with other editing models (Yan et al., 2026; Liu et al., 2025c). The source image already determines most of the output's layout and appearance (Lu et al., 2025), so tokens outside the edited region can be committed together at little cost (Chen et al., 2025b). This leaves distillation less room to help.

Entropy-matched guidance helps the student follow the instruction. Without it, the student's entropy keeps falling over the decoding steps and ends at 1.55, below the 2.28 of the teacher's CFG target (Figure 2 and Table 18). Such an overconfident student tends to copy the source image instead of making the edit. With entropy-matched guidance, the final entropy stays at 3.32, and the I2I benchmark score rises from 2.04 to 2.31. This 13% gain is the largest in Table 21.

Omni-Diffusion-Distill leads on Add and Compose. On Add it scores 3.26, while the best other distilled method scores 2.52. It ranks second on the I2I benchmark Overall and Graph-200K. It is also above the few-step teacher on Compose and Adjust, although the teacher uses 24 NFE and the

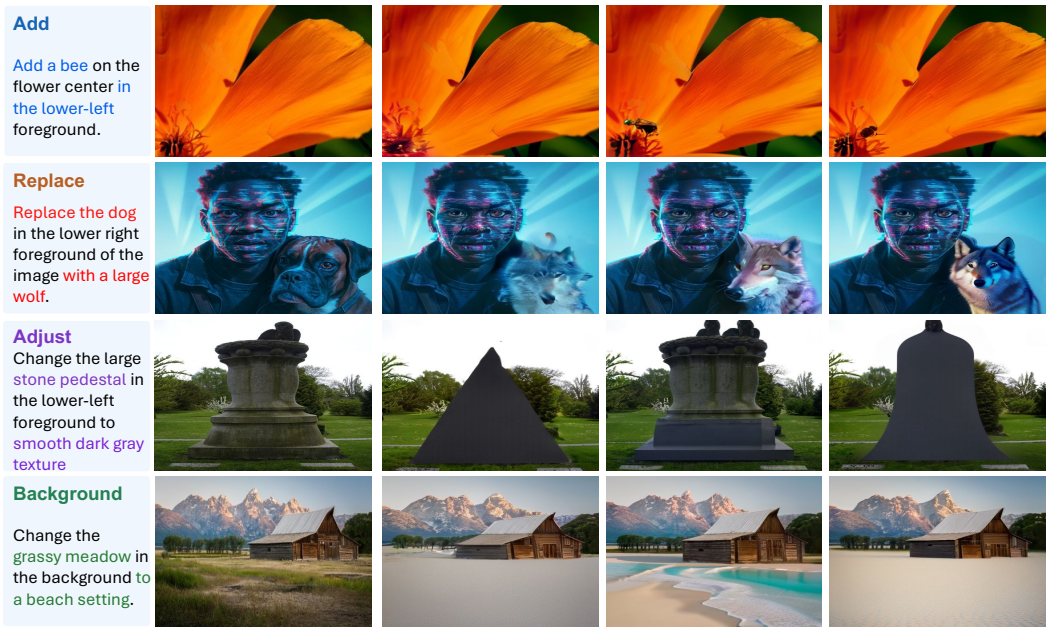


Figure 6: **Qualitative comparison on image editing.** We compare with the teacher and the strongest distilled baseline Di[M]O. Distilled models use 8 decoding steps, and the teacher uses 128.

Table 2: **Quantitative Results on I2I. The few-step teacher is already strong, so distillation has less room to help. Omni-Diffusion-Distill is on par with the best distilled baseline.** I2I benchmark scores range from 1 to 5. Overall is the mean of nine edit types, and four are shown. Graph-200K reports DINOv2 image similarity.

Method	NFE ↓	I2I Benchmark					Graph-200K
		Add ↑	Compose ↑	Adjust ↑	Style ↑	Overall ↑	DINOv2 ↑
Teacher (full-step)	128×3	3.62	1.94	2.29	3.89	2.66	0.799
T3D	8×1	2.12	1.78	**2.49**	3.72	2.28	0.774
Di[M]O	8×1	2.52	1.85	2.16	**3.90**	**2.38**	**0.794**
CDLM	8×1	1.47	1.51	1.45	3.11	1.68	0.657
Omni-Diffusion-Distill	8×1	**3.26**	**1.90**	2.45	3.83	2.31	0.789
Teacher (few-step)	8×3	3.32	1.56	2.41	3.93	2.59	0.792

student uses 8. On Graph-200K it scores 0.789, while the full-step teacher scores 0.799 with 48× more forward passes. At 4 and 2 steps, it beats the few-step teacher on Graph-200K (Table 10).

4.4 MMU: Step Reduction Primarily Affects Open-Ended Generation

Table 3: **Quantitative Results on Discriminative MMU. Discriminative scores remain stable with fewer steps.** All benchmarks use their standard parser and metric.

Method	NFE ↓	POPE ↑	MME-P ↑	MMBench ↑	SEED ↑	MMMU ↑
Teacher (full-step)	512	86.8	1552	91.2	85.3	58.3
T3D	64	82.8	1429	60.2	78.6	37.8
Di[M]O	64	87.0	1458	57.0	79.5	41.2
CDLM	64	79.5	1344	78.4	79.2	50.7
Omni-Diffusion-Distill	64	**87.4**	**1537**	**89.4**	**84.6**	**56.3**
Teacher (few-step)	64	86.4	1552	90.8	85.1	58.2

We evaluate MMU with two groups of benchmarks. Discriminative benchmarks check a short answer, and open-ended benchmarks use a GPT judge on the whole output. Tables 3 and 4 report the two groups, and Appendix C.5 gives the judging protocols.

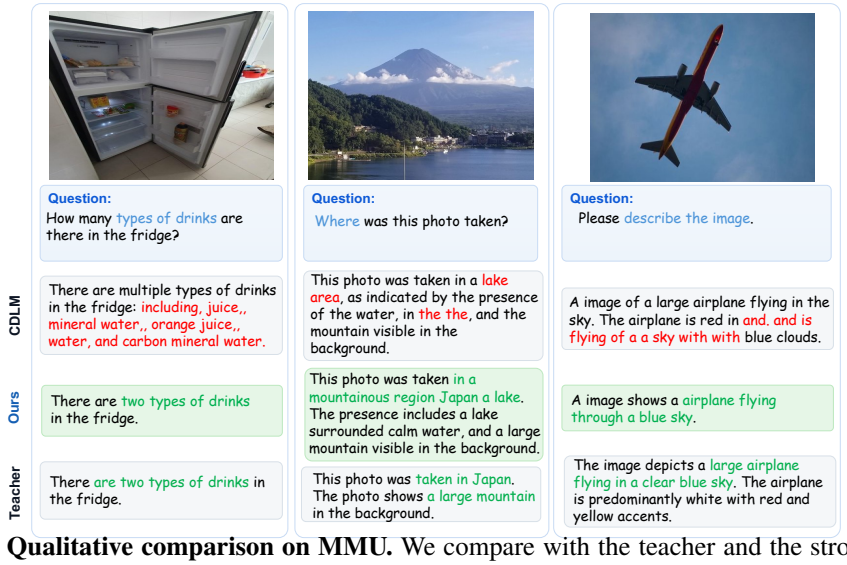


Figure 7: **Qualitative comparison on MMU.** We compare with the teacher and the strongest distilled baseline CDLM. Distilled models use 64 decoding steps, and the teacher uses 512. Red marks repeated or incorrect text, and green marks semantically correct content.

Table 4: **Quantitative Results on Open-Ended MMU.** A GPT judge scores the outputs. LLaVA-Bench reports a relative score, and DeCapBench reports DCSCORE.

Method	NFE ↓	MM-Vet ↑	Vibe-Eval ↑	LLaVA-Bench ↑	COCO ↑	DeCapBench ↑
Teacher (full-step)	512	21.3	12.36	25.1	67.2	44.5
T3D	64	15.3	3.81	14.7	21.3	13.8
Di[M]O	64	17.2	6.04	14.5	33.4	15.9
CDLM	64	18.5	7.71	15.6	50.3	16.9
Omni-Diffusion-Distill	64	**20.0**	**8.18**	**16.9**	**57.2**	**19.6**
Teacher (few-step)	64	15.5	5.30	14.7	28.4	16.2

Discriminative scores remain stable with fewer steps. When the teacher goes from 512 to 64 steps, all five scores stay within half a point. MME-P stays at 1552, and MMMU moves from 58.3 to 58.2. Omni-Diffusion-Distill is just as stable from 64 down to 16 steps. It is also the best distilled method on all five benchmarks.

Open-ended scores drop sharply with fewer steps, and Omni-Diffusion-Distill recovers much of this loss. These benchmarks check the whole answer, so repeated words and broken sentences lower the score. At 64 steps, the teacher loses more than half of its COCO captioning score and about two thirds of its DeCapBench score. Omni-Diffusion-Distill beats the few-step teacher on all five open-ended benchmarks. On COCO captioning it scores 57.2, while the few-step teacher scores 28.4. In Figure 7, CDLM repeats words and leaves phrases unfinished, and Table 18 measures this repetition. A gap to the full-step teacher remains, and it is largest on DeCapBench. Appendices D and E give more results.

5 Conclusion

We present Omni-Diffusion-Distill, a few-step distillation framework for fully discrete unified multimodal dLLMs. The student learns to skip decoding steps from cached teacher trajectories and is then refined on its own rollouts, with one on-policy loop for images and text. To address the challenge each modality faces, entropy-matched guidance keeps the image student from becoming overconfident, and a pairwise collision penalty reduces repetition in parallel text decoding. Distillation also makes the text attention block-causal, so understanding reuses its KV cache exactly. The student generates an image in 8 steps and answers a question in 64, 18.2× and 21.2× faster than the teacher, and outperforms all distilled baselines on most benchmarks, even approaching the full-step teacher on several. Our approach is limited to unified dLLMs that represent text and images as discrete tokens, and we validate it only on Lumina-DiMOO. The student still trails the full-step teacher, most on text rendering and detailed captioning. On image editing, it also trails the few-step teacher at 8 steps, although it uses a third of the NFE.

Ethics Statement

This work uses only publicly released models and datasets. It involves no human subjects, no new data collection and no personally identifying information.

Reproducibility Statement

Section 3 describes the method. Algorithm 1 and Appendix B give both training algorithms and the exact form of every loss. Appendix C.1 lists the training data and the teacher trajectory settings. Appendix C.2 gives the training configuration and the cost of each stage. Section 4.1 defines the decoding budgets and how NFE is counted. Appendix C.5 gives the scoring protocol of every open-ended benchmark, including the judge model, the evaluation subset and the protocol we defined for COCO captioning.

AI Use Statement

We used large language models in three ways. First, we used them to draft and polish the text of this paper. Second, we used them for retrieval and discovery, such as searching for related work. Every suggested reference was checked against its published source before we cited it. Third, we used them to assist research ideation and execution, such as discussing method designs and writing and debugging code. The authors made all research decisions and reviewed all model output. All reported numbers come from experiments run by the authors.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/5be69a584901a26c521c2b51e40a4c20-Paper-Conference.pdf.

Inclusion AI, Tiwei Bie, Haoxing Chen, Tieyuan Chen, Zhenglin Cheng, Long Cui, Kai Gan, Zhicheng Huang, Zhenzhong Lan, Haoquan Li, et al. Llada2. 0-uni: Unifying multimodal understanding and generation with diffusion large language model. *arXiv preprint arXiv:2604.20796*, 2026.

Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2503.09573.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/958c530554f78bcd8e97125b70e6973d-Paper.pdf.

Yara Bahram, Mélodie Desbos, Mohammadhadi Shateri, and Eric Granger. Uni-DAD: Unified distillation and adaptation of diffusion models for few-step few-shot image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2511.18281.

Victor Besnier, Mickael Chen, David Hurych, Eduardo Valle, and Matthieu Cord. Halton scheduler for masked generative image transformer. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2503.17076.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402. IEEE, 2023.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2205.14987.

Jingjing Chang, Yixiao Fang, Peng Xing, Shuhan Wu, Wei Cheng, Rui Wang, Xianfang Zeng, Gang Yu, and Hai-Bao Chen. OneIG-Bench: Omni-dimensional nuanced evaluation for image generation. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025.

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, Le Xue, Caiming Xiong, and Ran Xu. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *ArXiv*, abs/2505.09568, 2025a. URL https://api.semanticscholar.org/CorpusID:278602689.

Pengtao Chen, Xianfang Zeng, Maosen Zhao, Mingzhu Shen, Peng Ye, Bangyin Xiang, Zhibo Wang, Wei Cheng, Gang Yu, and Tao Chen. RegionE: Adaptive region-aware generation for efficient image editing. *arXiv preprint arXiv:2510.25590*, 2025b. URL https://arxiv.org/abs/2510.25590.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025c.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

Zigeng Chen, Gongfan Fang, Xinyin Ma, Ruonan Yu, and Xinchao Wang. dParallel: Learnable parallel decoding for dLLMs. *arXiv preprint arXiv:2509.26488*, 2025d. URL https://arxiv.org/abs/2509.26488.

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Self-Forcing++: Towards minute-scale high-quality video generation. *arXiv preprint arXiv:2510.02283*, 2025. URL https://arxiv.org/abs/2510.02283.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Justin Deschenaux and Caglar Gulcehre. Beyond autoregression: Fast LLMs via self-distillation through time. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.21035.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. URL https://arxiv.org/abs/2012.09841.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. MME: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. URL https://arxiv.org/abs/2306.13394.

Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. *arXiv preprint arXiv:2310.11513*, 2023. URL https://arxiv.org/abs/2310.11513.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. DiffuSeq: Sequence to sequence text generation with diffusion models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.08933.

Jiawei Guo, Tianyu Zheng, Yizhi Li, Yuelin Bai, Bo Li, Yubo Wang, King Zhu, Graham Neubig, Wenhu Chen, and Xiang Yue. MAmmoTH-VL: Eliciting multimodal reasoning with instruction tuning at scale. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 13869–13920, 2025.

Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. SSD-LM: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023. URL https://arxiv.org/abs/2210.17432.

Satoshi Hayakawa, Yuhta Takida, Masaaki Imaizumi, Hiromi Wakaki, and Yuki Mitsufuji. Distillation of discrete diffusion through dimensional correlations. In *International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2410.08709.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL https://arxiv.org/abs/2207.12598.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 12454–12465. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/67d96d458abdef21792e6d8e590244e7-Paper.pdf.

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. URL https://arxiv.org/abs/2403.05135. Introduces DPG-Bench.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2506.08009.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2603.07079.

Wonjun Kang, Kevin Galim, Seunghyuk Oh, Minjae Lee, Yuchen Zeng, Shuibai Zhang, Coleman Hooper, Yuezhou Hu, Hyung Koo, Nam Ik Cho, and Kangwook Lee. Parallelbench: Understanding the trade-offs of parallel decoding in diffusion llms. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 89900–89938, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/9156b0f6dfa9bbd18c79cc459ef5d61c-Paper-Conference.pdf.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.02279.

Minseo Kim, Chenfeng Xu, Coleman Hooper, Harman Singh, Ben Athiwaratkun, Ce Zhang, Kurt Keutzer, and Amir Gholami. CDLM: Consistency diffusion language models for faster sampling. *arXiv preprint arXiv:2511.19269*, 2025. URL https://arxiv.org/abs/2511.19269.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2402.03898.

Jonas Kohler, Albert Pumarola, Edgar Schönfeld, Artsiom Sanakoyeu, Roshan Sumbaly, Peter Vajda, and Ali Thabet. Imagine flash: Accelerating Emu diffusion models with backward distillation. *arXiv preprint arXiv:2405.05224*, 2024. URL https://arxiv.org/abs/2405.05224.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. Seed-bench: Benchmarking multimodal large language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13299–13308, 2024. doi: 10.1109/CVPR52733.2024.01263.

Lijiang Li, Zuwei Long, Yunhang Shen, Heting Gao, Haoyu Cao, Xing Sun, Caifeng Shan, Ran He, and Chaoyou Fu. Omni-diffusion: Unified multimodal understanding and generation with masked discrete diffusion. *arXiv preprint arXiv:2603.06577*, 2026a.

Pengxiang Li, Yefan Zhou, Dilxat Muhtar, Lu Yin, Shilin Yan, Li Shen, Soroush Vosoughi, and Shiwei Liu. Diffusion language models know the answer before decoding. *arXiv preprint arXiv:2508.19982*, 2025a. URL https://arxiv.org/abs/2508.19982.

Pengyu Li, Zhitao Gao, Lingling Zhang, Muye Huang, Yuanming Li, Zesheng Yang, Fangzhi Xu, and Jun Liu. Visual-OPSD: Cross-modal on-policy self-distillation for efficient unified multimodal reasoning. *arXiv preprint arXiv:2606.18974*, 2026b. URL https://arxiv.org/abs/2606.18974.

Quanhao Li, Junqiu Yu, Kaixun Jiang, Yujie Wei, Zhen Xing, Pandeng Li, Ruihang Chu, Shiwei Zhang, Yu Liu, and Zuxuan Wu. DiffusionOPD: A unified perspective of on-policy distillation in diffusion models. *arXiv preprint arXiv:2605.15055*, 2026c. URL https://arxiv.org/abs/2605.15055.

Shufan Li, Konstantinos Kallidromitis, Hritik Bansal, Akash Gokul, Yusuke Kato, Kazuki Kozuka, Jason Kuen, Zhe Lin, Kai-Wei Chang, and Aditya Grover. Lavida: A large diffusion language model for multimodal understanding. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 105101–105134. Curran Associates, Inc., 2025b. doi: 10.52202/085713-3511. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/975affbe7b5f3b55fba62247b6877b1c-Paper-Conference.pdf.

Shufan Li, Jiuxiang Gu, Kangning Liu, Zhe Lin, Zijun Wei, Aditya Grover, and Jason Kuen. Lavida-o: Elastic large masked diffusion models for unified multimodal understanding and generation. In *International Conference on Learning Representations*, volume 2026, pp. 50571–50603, 2026d.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-LM improves controllable text generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2205.14217.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026e. URL https://arxiv.org/abs/2604.13016.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. URL https://arxiv.org/abs/2305.10355. POPE.

Zhong-Yu Li, Ruoyi Du, Juncheng Yan, Le Zhuo, Zhen Li, Peng Gao, Zhanyu Ma, and Ming-Ming Cheng. Visualcloze: A universal image generation framework via visual in-context learning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18969–18979, 2025c. doi: 10.1109/ICCV51701.2025.01763.

Zijie Li, Henry Li, Yichun Shi, Amir Barati Farimani, Yuval Kluger, Linjie Yang, and Peng Wang. Dual diffusion for unified image generation and understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2779–2790. IEEE, 2025d.

Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, Yatian Pang, and Li Yuan. UniWorld-V1: High-resolution semantic encoders for unified visual understanding and generation. *arXiv preprint arXiv:2506.03147*, 2025.

Zhenghao Lin, Yeyun Gong, Yelong Shen, Tong Wu, Zhihao Fan, Chen Lin, Nan Duan, and Weizhu Chen. Text generation with diffusion language models: A pre-training approach with continuous paragraph denoise. In *International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2212.11685.

Anji Liu, Oliver Broadrick, Mathias Niepert, and Guy Van den Broeck. Discrete copula diffusion. In *International Conference on Learning Representations (ICLR)*, 2025a. URL https://arxiv.org/abs/2410.01949.

Enshu Liu, Xuefei Ning, Yu Wang, and Zinan Lin. Distilled decoding 1: One-step sampling of image auto-regressive models with flow matching. In *International Conference on Learning Representations (ICLR)*, 2025b. URL https://arxiv.org/abs/2412.17153.

Fangbing Liu, Pengfei Duan, Wen Li, and Yi He. LGCC: Enhancing flow matching based text-guided image editing with local Gaussian coupling and context consistency. *arXiv preprint arXiv:2511.01894*, 2025c. URL https://arxiv.org/abs/2511.01894.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023a.

Junpeng Liu, Tianyue Ou, Yifan Song, Yuxiao Qu, Wai Lam, Chenyan Xiong, Wenhu Chen, Graham Neubig, and Xiang Yue. Harnessing webpage UIs for text-rich visual understanding. In *International Conference on Learning Representations (ICLR)*, 2025d.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023b. URL https://arxiv.org/abs/2209.03003.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. InstaFlow: One step is enough for high-quality diffusion-based text-to-image generation. In *International Conference on Learning Representations (ICLR)*, 2024a. URL https://arxiv.org/abs/2309.06380.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. MMBench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision (ECCV)*, 2024b. URL https://arxiv.org/abs/2307.06281.

Zhiyuan Liu, Yicun Yang, Yaojie Zhang, Junjie Chen, Chang Zou, Qingyan Wei, Shaobo Wang, Yichen Zhu, and Linfeng Zhang. dLLM-Cache: Accelerating diffusion large language models with adaptive caching. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2506.06295.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2310.16834.

Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision, language, audio, and action. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26429–26445. IEEE, 2024.

Yanzuo Lu, Xin Xia, Manlin Zhang, Huafeng Kuang, Jianbin Zheng, Yuxi Ren, and Xuefeng Xiao. Hyper-Bagel: A unified acceleration framework for multimodal understanding and generation. *arXiv preprint arXiv:2509.18824*, 2025. URL https://arxiv.org/abs/2509.18824.

Run Luo, Xiaobo Xia, Lu Wang, Longze Chen, Renke Shan, Jing Luo, Min Yang, and Tat-Seng Chua. Next-omni: Towards any-to-any omnimodal foundation models with discrete flow matching. In *International Conference on Learning Representations*, volume 2026, pp. 147298–147334, 2026.

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. URL https://arxiv.org/abs/2310.04378.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik P. Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. URL https://arxiv.org/abs/2210.03142.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan LI. Large language diffusion models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 50608–50646. Curran Associates, Inc., 2025. doi: 10.52202/085713-1689. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/48b383b24230e0e6e649d9c98dae4d8c-Paper-Conference.pdf.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2406.03736.

Piotr Padlewski, Max Bain, Matthew Henderson, Zhongkai Zhu, Nishant Relan, Hai Pham, Donovan Ong, Kaloyan Aleksiev, Aitor Ormazabal, Samuel Phua, Ethan Yeo, Eugenie Lamprecht, Qi Liu, Yuqi Wang, Eric Chen, Deyu Fu, Lei Li, Che Zheng, Cyprien de Masson d'Autume, Dani Yogatama, Mikel Artetxe, and Yi Tay. Vibe-Eval: A hard evaluation suite for measuring progress of multimodal language models. *arXiv preprint arXiv:2405.02287*, 2024.

Elia Peruzzo, Aritra Bhowmik, Guillaume Sautiere, Yuki M. Asano, and Amirhossein Habibian. Knowledge distillation for visual autoregressive models. *arXiv preprint arXiv:2606.06078*, 2026. URL https://arxiv.org/abs/2606.06078.

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. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741. Curran Associates, Inc., 2023. doi: 10.52202/075280-2338. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2406.07524.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2202.00512.

Guillaume Sanchez, Honglu Fan, Alexander Spangher, Elad Levi, Pawan Sasanka Ammanamanchi, and Stella Biderman. Stay on topic with classifier-free guidance. *arXiv preprint arXiv:2306.17806*, 2023.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023. URL https://arxiv.org/abs/2311.17042.

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. *arXiv preprint arXiv:2403.12015*, 2024. URL https://arxiv.org/abs/2403.12015.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2406.04329.

Qingyu Shi, Jinbin Bai, Zhuoran Zhao, Wenhao Chai, Kaidong Yu, Jianzong Wu, Yunhai Tong, Xiangtai Li, Xuelong Li, and Shuicheng Yan. Muddit: Liberating generation beyond text-to-image with a unified discrete diffusion model. *arXiv preprint arXiv:2505.23606*, 2025.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026. URL https://arxiv.org/abs/2604.00626.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2303.01469.

Yuerong Song, Xiaoran Liu, Ruixiao Li, Zhigeng Liu, Zengfeng Huang, Qipeng Guo, Ziwei He, and Xipeng Qiu. Sparse-dLLM: Accelerating diffusion LLMs with dynamic cache eviction. *arXiv preprint arXiv:2508.02558*, 2025. URL https://arxiv.org/abs/2508.02558.

Xingyu Su, Jacob Helwig, Shubham Parashar, Atharv Chagi, Lakshmi Jotsna, Degui Zhi, James Caverlee, Dileep Kalathil, and Shuiwang Ji. Data-efficient autoregressive-to-diffusion language models via on-policy distillation. *arXiv preprint arXiv:2606.06712*, 2026. URL https://arxiv.org/abs/2606.06712.

Pranav Subbaraman, Shufan Li, Siyan Zhao, and Aditya Grover. Accelerating inference of masked image generators via reinforcement learning. *arXiv preprint arXiv:2512.01094*, 2025. URL https://arxiv.org/abs/2512.01094.

Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Emu: Generative pretraining in multimodality. In *International conference on learning representations*, volume 2024, pp. 12352–12380, 2024.

Onkar Susladkar, Tushar Prakash, Gayatri Deshmukh, Kiet A Nguyen, Jiaxun Zhang, Adheesh Juvekar, Tianshu Bao, Lin Chai, Sparsh Mittal, Inderjit S Dhillon, et al. Best of both worlds: Multimodal reasoning and generation via unified discrete flow matching. *arXiv preprint arXiv:2602.12221*, 2026.

Alexander Swerdlow, Mihir Prabhudesai, Siddharth Gandhi, Deepak Pathak, and Katerina Fragkiadaki. Unified multimodal discrete diffusion. *arXiv preprint arXiv:2503.20853*, 2025.

Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. OminiControl: Minimal and universal control for diffusion transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Yao Teng, Han Shi, Xian Liu, Xuefei Ning, Guohao Dai, Yu Wang, Zhenguo Li, and Xihui Liu. Accelerating auto-regressive text-to-image generation with training-free speculative Jacobi decoding. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.01699.

Ye Tian, Ling Yang, Jiongfan Yang, Anran Wang, Yu Tian, Haochen Wang, Zhiyang Teng, Zhuochen Wang, Yinjie Wang, Yunhai Tong, et al. Mmada-parallel: Multimodal large diffusion language models for thinking-aware editing and generation. In *International Conference on Learning Representations*, volume 2026, pp. 36089–36123, 2026.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL https://arxiv.org/abs/1711.00937.

Chaoyang Wang and Yunhai Tong. One-step distillation of discrete diffusion image generators via fixed-point iteration. *arXiv preprint arXiv:2605.21484*, 2026. URL https://arxiv.org/abs/2605.21484.

Jin Wang, Yao Lai, Aoxue Li, Shifeng Zhang, Jiacheng Sun, Ning Kang, Chengyue Wu, Zhenguo Li, and Ping Luo. Fudoki: Discrete flow-based unified understanding and generation via kinetic-optimal velocities. *Advances in Neural Information Processing Systems*, 38:81402–81440, 2026a.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

Xu Wang, Chenkai Xu, Yijie Jin, Jiachun Jin, Hao Zhang, Kai Yu, and Zhijie Deng. Diffusion llms can do faster-than-ar inference via discrete diffusion forcing. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 2282–2299, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/04bb76a98d9f32226b3beba7bd26a51f-Paper-Conference.pdf.

Yibin Wang, Zhimin Li, Yuhang Zang, Jiazi Bu, Yujie Zhou, Yi Xin, Junjun He, Chunyu Wang, Qinglin Lu, Cheng Jin, and Jiaqi Wang. UniGenBench++: A unified semantic evaluation benchmark for text-to-image generation. *arXiv preprint arXiv:2510.18701*, 2025a.

Yibin Wang, Zhimin Li, Yuhang Zang, Yujie Zhou, Jiazi Bu, Chunyu Wang, Qinglin Lu, Cheng Jin, and Jiaqi Wang. Pref-GRPO: Pairwise preference reward-based GRPO for stable text-to-image reinforcement learning. *arXiv preprint arXiv:2508.20751*, 2025b.

Qingyan Wei, Yaojie Zhang, Zhiyuan Liu, Puyu Zeng, Yuxuan Wang, Biqing Qi, Dongrui Liu, and Linfeng Zhang. Accelerating diffusion large language models with SlowFast sampling: The three golden principles. *arXiv preprint arXiv:2506.10848*, 2025a. URL https://arxiv.org/abs/2506.10848.

Xinyu Wei, Jinrui Zhang, Zeqing Wang, Hongyang Wei, Zhen Guo, Bairui Li, and Lei Zhang. TIIF-Bench: How does your T2I model follow your instructions? *arXiv preprint arXiv:2506.02161*, 2025b.

Chen-Yuan Wu, Peng-Fei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ze Liu, Ziyi Xia, Chaofan Li, Haoge Deng, Jia-Hao Wang, Kun Luo, Bo Zhang, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. Omnigen2: Towards instruction-aligned multimodal generation. 2025a. URL https://api.semanticscholar.org/CorpusID:279999713.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12966–12977. IEEE, 2025b.

Chengyue Wu, Hao Zhang, Shuchen Xue, Shizhe Diao, Yonggan Fu, Zhijian Liu, Pavlo Molchanov, Ping Luo, Song Han, and Enze Xie. Fast-dLLM v2: Efficient block-diffusion LLM. *arXiv preprint arXiv:2509.26328*, 2025c. URL https://arxiv.org/abs/2509.26328.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dLLM: Training-free acceleration of diffusion LLM by enabling KV cache and parallel decoding. *arXiv preprint arXiv:2505.22618*, 2025d. URL https://arxiv.org/abs/2505.22618.

Tong Wu, Zhihao Fan, Xiao Liu, Hai-Tao Zheng, Yeyun Gong, yelong shen, Jian Jiao, Juntao Li, zhongyu wei, Jian Guo, Nan Duan, and Weizhu Chen. Ar-diffusion: Auto-regressive diffusion model for text generation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 39957–39974. Curran Associates, Inc., 2023. doi: 10.52202/075280-1737. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/7d866abba506e5a56335e4644ebe18f9-Paper-Conference.pdf.

Yi Xin, Qi Qin, Siqi Luo, Kaiwen Zhu, Juncheng Yan, Yan Tai, Jiayi Lei, Yuewen Cao, Keqi Wang, Yibin Wang, et al. Lumina-dimoo: An omni diffusion large language model for multi-modal generation and understanding. *arXiv preprint arXiv:2510.06308*, 2025.

Chenkai Xu, Xu Wang, Zhenyi Liao, Yishun Li, TianQi Hou, and Zhijie Deng. UniCM: A unified consistency model for efficient multimodal generation and understanding. In Maria

Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 18214–18233, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.906. URL https://aclanthology.org/2026.findings-acl.906/.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2304.05977.

Zhengan Yan, Shikang Zheng, Haoran Qin, Xiaobing Tu, Yinggui Wang, Jiacheng Liu, Jiaxuan Ren, Yuqi Lin, Peiliang Cai, Jinkui Ren, Xiantao Zhang, and Linfeng Zhang. SpecEdit: Training-free acceleration for diffusion based image editing via semantic locking. *arXiv preprint arXiv:2605.02152*, 2026. URL https://arxiv.org/abs/2605.02152.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. *Advances in Neural Information Processing Systems*, 38:138867–138907, 2026.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *ArXiv*, abs/2508.15487, 2025a. URL https://api.semanticscholar.org/CorpusID:280700361.

Qinghao Ye, Xianhan Zeng, Fu Li, Chunyuan Li, and Haoqi Fan. Painting with words: Elevating detailed image captioning with benchmark and alignment learning. In *International Conference on Learning Representations (ICLR)*, 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a. URL https://arxiv.org/abs/2405.14867.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b. URL https://arxiv.org/abs/2311.18828.

Jaehoon Yoo, Wonjung Kim, Chanhyuk Lee, and Seunghoon Hong. Infinite mask diffusion for few-step distillation. *arXiv preprint arXiv:2605.10518*, 2026. URL https://arxiv.org/abs/2605.10518.

Zebin You, Jingyang Ou, Xiaolu Zhang, Jun Hu, Jun Zhou, and Chongxuan Li. Effective and efficient masked image generation models. *arXiv preprint arXiv:2503.07197*, 2025. URL https://arxiv.org/abs/2503.07197.

Zebin You, Shen Nie, Xiaolu Zhang, JUN ZHOU, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. Llada-v: Large language diffusion models with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10093–10105, June 2026a.

Zebin You, Xiaolu Zhang, Jun Zhou, Chongxuan Li, and Ji-Rong Wen. Llada-o: An effective and length-adaptive omni diffusion model. *arXiv preprint arXiv:2603.01068*, 2026b.

Runpeng Yu, Xinyin Ma, and Xinchao Wang. Dimple: Discrete diffusion multimodal large language model with parallel decoding. *ArXiv*, abs/2505.16990, 2025. URL https://api.semanticscholar.org/CorpusID:278789456.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. MM-Vet: Evaluating large multimodal models for integrated capabilities. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. URL https://arxiv.org/abs/2311.16502.

Bowen Zhang. A formula-driven survey and research agenda for on-policy distillation. *arXiv preprint arXiv:2606.22793*, 2026. URL https://arxiv.org/abs/2606.22793.

Tunyu Zhang, Xinxi Zhang, Ligong Han, Haizhou Shi, Xiaoxiao He, Zhuowei Li, Hao Wang, Kai Xu, Akash Srivastava, Chengzhi Mao, Hao Wang, Vladimir Pavlovic, and Dimitris N. Metaxas. Few-step diffusion language models via trajectory self-distillation. *arXiv preprint arXiv:2602.12262*, 2026. URL https://arxiv.org/abs/2602.12262.

Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. UltraEdit: Instruction-based fine-grained image editing at scale. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026. URL https://arxiv.org/abs/2601.18734.

Haoyang Zheng, Xinyang Liu, Cindy Xiangrui Kong, Nan Jiang, Zheyuan Hu, Weijian Luo, Wei Deng, and Guang Lin. Ultra-fast language generation via discrete diffusion divergence instruct. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2509.25035.

Kaiwen Zheng, Yongxin Chen, Huayu Chen, Guande He, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Direct discriminative optimization: Your likelihood-based visual generative model is secretly a GAN discriminator. In *International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2503.01103. ICML 2025 Spotlight.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. In *International Conference on Learning Representations*, volume 2025, pp. 6446–6469, 2025.

Haoyang Zhou, Li Kong, Shijie Ren, Xiting Wang, Shuang Liang, Guowei Wang, and Zhenxuan Pan. TAD: Temporal-aware trajectory self-distillation for fast and accurate diffusion LLM. *arXiv preprint arXiv:2605.09536*, 2026. URL https://arxiv.org/abs/2605.09536.

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 *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2404.04057.

Yuanzhi Zhu, Xi Wang, Stéphane Lathuilière, and Vicky Kalogeiton. Di[M]O: Distilling masked diffusion models into one-step generator. *arXiv preprint arXiv:2503.15457*, 2025a. URL https://arxiv.org/abs/2503.15457. ICCV 2025.

Yuanzhi Zhu, Xi Wang, Stéphane Lathuilière, and Vicky Kalogeiton. Soft-Di[M]O: Improving one-step discrete image generation with soft embeddings. *arXiv preprint arXiv:2509.22925*, 2025b. URL https://arxiv.org/abs/2509.22925.

A Extended Related Work

A.1 Unified Multimodal Understanding and Generation

Unified models differ mainly in how they represent and generate images alongside text. Autoregressive models generate image tokens with the same transformer that generates text (Team, 2024; Wang et al., 2024; Lu et al., 2024). Janus and Janus-Pro keep this design but use separate visual encoders for understanding and generation (Wu et al., 2025b; Chen et al., 2025c). Hybrid designs keep text autoregressive but generate images with continuous diffusion, either inside the same transformer or through a separate decoder or branch (Zhou et al., 2025; Sun et al., 2024; Deng et al., 2025; Wu et al., 2025a).

A third family generates with an iterative discrete denoiser. One group uses masked diffusion over a shared text and image vocabulary for both understanding and generation. It includes MMaDA and its parallel variant (Yang et al., 2026; Tian et al., 2026), Lumina-DiMOO (Xin et al., 2025), UniDisc (Swerdlow et al., 2025), Muddit (Shi et al., 2025) and Omni-Diffusion (Li et al., 2026a). Lavida-O also denoises VQ image tokens with masked diffusion, but processes them with a separate and smaller generation branch (Li et al., 2026d). Unified LLaDA variants keep a discrete text denoiser but attach a continuous image decoder (AI et al., 2026; You et al., 2026b). A closely related group replaces masked diffusion with discrete flow matching and keeps the same iterative denoising formulation (Wang et al., 2026a; Luo et al., 2026; Susladkar et al., 2026). D-DiT combines discrete text diffusion with continuous image diffusion in one bidirectional model (Li et al., 2025d). A separate line uses dLLM decoding only for understanding. These models pair a masked diffusion text decoder with a vision encoder and do not generate images (You et al., 2026a; Li et al., 2025b; Yu et al., 2025).

A.2 Diffusion Large Language Models and Discrete Generative Modeling

Discrete diffusion extends denoising to categorical data. Early work defines the forward corruption as a discrete Markov process with uniform, absorbing or structured kernels (Austin et al., 2021; Hoogeboom et al., 2021), and continuous-time formulations give tractable objectives and samplers (Campbell et al., 2022). The absorbing kernel, also called the mask kernel, works best for language. Score entropy (Lou et al., 2024), simplified masked diffusion ELBOs (Sahoo et al., 2024; Shi et al., 2024) and reparameterizations that view masked diffusion as modeling clean-data conditionals (Ou et al., 2025) steadily closed the gap to autoregressive models. This line led to dLLMs at LLM scale (Nie et al., 2025; Ye et al., 2025a). Another thread interpolates between autoregression and diffusion, with block-wise (Arriola et al., 2025), semi-autoregressive (Han et al., 2023) or position-dependent (Wu et al., 2023) decoding, or with continuous diffusion over embeddings (Li et al., 2022; Gong et al., 2023; Lin et al., 2023). A key property of dLLMs is that they predict an explicit distribution at each position. In continuous diffusion, the distribution of the generator is implicit.

A.3 Distillation for Few-Step Generation

Few-step distillation has developed along three largely separate lines, for image generators, autoregressive LLMs and dLLMs. We review them in turn, and then discuss unified models and on-policy distillation. Table 5 compares representative methods by whether training is on-policy, whether the target is a fixed teacher or a snapshot of the student, whether the generator is discrete, and the modality.

Distilling image generators. Few-step distillation of continuous image diffusion models is well studied. Trajectory-based methods distill several teacher steps into one. Examples include progressive distillation (Salimans & Ho, 2022), guidance distillation (Meng et al., 2023), consistency and trajectory models (Song et al., 2023; Kim et al., 2024; Luo et al., 2023), and rectified flow (Liu et al., 2023b; 2024a). Distribution matching and score-based methods instead match the output distributions of the student and the teacher. They usually need a learned auxiliary network, because the score of the continuous student cannot be evaluated directly (Yin et al., 2024b;a; Zhou et al., 2024). Uni-DAD combines distribution matching distillation with few-shot domain adaptation in a single stage (Bahram et al., 2026). Adversarial variants add a discriminator to improve sharpness at one to four steps (Sauer et al., 2023; 2024). On-policy variants train the student on its own generations to

Table 5: **Representative Methods for Few-Step and Accelerated Generation.** Methods are grouped by model class. Policy: whether the student is trained on its own generations. Target/ref.: T is a fixed teacher, and S is a snapshot or self-reference of the student. Disc.: discrete-token generator. Modality: I is image, T is text, and M is unified multimodal. A dash marks methods that are not distillation objectives, so policy and target do not apply.

Method	Family	Policy	Target/ref.	Disc.	Modality
Image generators					
Progressive distillation (Salimans & Ho, 2022)	trajectory	off	T (iterated)		I
Consistency models (Song et al., 2023)	trajectory	off	T+S		I
DMD (Yin et al., 2024b)	dist.-match	on	T (+auxiliary network)		I
DMD2 (Yin et al., 2024a)	dist.-match + adv.	on	T (+auxiliary network)		I
DiffusionOPD (Li et al., 2026c)	on-policy KD	on	T		I
Di[M]O (Zhu et al., 2025a)	dist.-match	on	T (+auxiliary network)	✓	I
Di4C (Hayakawa et al., 2025)	trajectory	off	T+S	✓	I, T
Autoregressive LLMs					
GKD (Agarwal et al., 2024)	on-policy KD	on	T	✓	T
dLLMs					
CDLM (Kim et al., 2025)	trajectory	off	T+S	✓	T
SDTT (Deschenaux & Gulcehre, 2025)	trajectory	off	T (round-refreshed)	✓	T
T3D (Zhang et al., 2026)	trajectory + pref.	off	T + S (round-refreshed)	✓	T
DiDi-Instruct (Zheng et al., 2026)	dist.-match + adv.	on	T (+auxiliary network)	✓	T
AR→Diff. OPD (Su et al., 2026)	on-policy KD	on	T	✓	T
D2F (Wang et al., 2026b)	block-AR distill.	off	T	✓	T
Infinite-mask (Yoo et al., 2026)	mask parameterization	—	—	✓	T
Fast-dLLM (Wu et al., 2025d)	training-free cache + parallel decoding	—	—	✓	T
Unified multimodal (understanding + generation)					
UniCMs (Xu et al., 2026)	trajectory	off	T+S	✓	M
Hyper-Bagel (Lu et al., 2025)	multi-stage	mixed	T+S (+auxiliary network)		M
Omni-Diffusion-Distill	**dist.-match + on-policy KD + pref.**	**off→on**	**T + S (refreshed)**	✓	**M**

remove the mismatch between training and inference (Yin et al., 2024a; Kohler et al., 2024; Huang et al., 2025; Cui et al., 2025). Distillation for discrete image generators, which are masked or autoregressive models over VQ codes, is less studied but growing. Di[M]O distills a masked image generator into a one-step sampler with token-level distribution matching on re-masked outputs. It keeps entropy in the student by randomizing the token initialization (Zhu et al., 2025a), and its soft-embedding variant further improves one-step quality (Zhu et al., 2025b). Other work distills discrete diffusion image generators through dimensional correlations (Hayakawa et al., 2025) or fixed-point iteration (Wang & Tong, 2026). For autoregressive image models, one-step samplers have been obtained with flow matching (Liu et al., 2025b), and knowledge distillation has been applied to visual autoregressive models (Peruzzo et al., 2026). Apart from distillation, other methods speed up masked and autoregressive image sampling with better unmasking schedules (Besnier et al., 2025; You et al., 2025), speculative Jacobi decoding (Teng et al., 2025) or reward-tuned step reduction (Subbaraman et al., 2025).

Distilling language models. For autoregressive LLMs, classical knowledge distillation matches teacher logits (Hinton et al., 2015). Off-policy matching suffers from exposure bias, because the student's own generations drift away from the teacher's. On-policy methods address this by training on sequences generated by the student, as in generalized KD, which supports divergences such as the generalized Jensen-Shannon divergence (Agarwal et al., 2024). DistiLLM combines a skew KL objective with an adaptive off-policy scheme that reuses student-generated sequences (Ko et al., 2024). Another line uses preference objectives such as direct preference optimization (Rafailov et al., 2023), which pushes the model toward preferred responses and away from dispreferred ones. Direct discriminative optimization carries this parameterization to visual generative models and contrasts real data with samples from a frozen reference model (Zheng et al., 2025).

Distilling diffusion large language models. Distillation of dLLMs is recent. For text, consistency and trajectory-based methods distill dLLMs to a few steps (Kim et al., 2025; Deschenaux & Gulcehre, 2025; Zhang et al., 2026; Zhou et al., 2026), and a discrete distribution matching objective adapts Diff-Instruct to dLLMs (Zheng et al., 2026). Few-step dLLMs also make errors because they commit positions independently. Infinite-state masking reduces this error (Yoo et al., 2026), confidence thresholds address it at inference (Wu et al., 2025d), and dimensional-correlation distillation targets it during training (Hayakawa et al., 2025). Many other works accelerate dLLM inference

without full distillation. Training-free methods use KV caching and parallel decoding (Wu et al., 2025d), or adaptive sampling and cache eviction (Wei et al., 2025a; Li et al., 2025a; Liu et al., 2026; Song et al., 2025). Trained methods shorten the schedule through certainty forcing, block diffusion fine-tuning, or distillation into a block-autoregressive hybrid (Chen et al., 2025d; Wu et al., 2025c; Wang et al., 2026b). Each of these methods targets a single modality.

Distilling unified multimodal models. Closest to us are the few works that distill or accelerate a unified model for understanding and generation. UniCMs (Xu et al., 2026) applies consistency distillation to Show-o on both modalities, but its text side follows an autoregressive parallel decoding trace instead of a diffusion process. Hyper-Bagel (Lu et al., 2025) accelerates BAGEL with speculative decoding and a multi-stage image pipeline. The first stage distills classifier-free guidance, and the image generator is a continuous diffusion model. Some unified dLLMs build few-step distillation into their generation decoder (AI et al., 2026). Others distill a unified teacher into an understanding-only student (Li et al., 2026b).

On-policy distillation. On-policy distillation (OPD) trains the student on its own generations instead of teacher-forced data (Agarwal et al., 2024; Song & Zheng, 2026; Zhang, 2026). It has developed rapidly for language models. Recent work improves stability by mixing forward and reverse KL according to teacher entropy (Jin et al., 2026), analyzes when and why OPD helps (Li et al., 2026e), and extends OPD to dLLMs (Su et al., 2026). In the last work, a block-causal student is trained on its own rollouts and supervised by the frozen autoregressive model it was converted from. Trajectory-based distillation for dLLMs instead supervises the student on states from the teacher's own rollouts (Zhang et al., 2026). Another thread makes the student its own teacher through a self-reference (Zhao et al., 2026). Li et al. (2026c) distill a student along its own rollouts with a closed-form KL at each step of the sampler trajectory. They cover several continuous image diffusion tasks, each with its own teacher. Existing OPD methods each handle one modality.

Omni-Diffusion-Distill differs from these works in two ways. First, both modalities are masked tokens of one model. A single on-policy framework therefore covers both, with a modality-specific state, teacher prediction and matching term. Second, a pairwise collision penalty for parallel text decoding and entropy-matched guidance for images address the degradation that step reduction causes in each modality.

B Algorithms and Loss Details

Algorithm 2 Offline Teacher Trajectory Collection

Input: dataset $\mathcal{O}$ of prompt-output pairs, teacher p_ϕ, number of teacher decoding steps T, keyframe steps $\mathcal{F} \subseteq \{1, \dots, T\}$, number of kept probabilities R
Output: cache $\mathcal{D}$. Each entry stores the prompt and output, the decoding step at which each output position is committed, and one record per keyframe. Each record holds the keyframe state, together with the top-R teacher log-probabilities and the residual probability at the positions committed since the previous keyframe
$\mathcal{D} \leftarrow \emptyset$
for each pair in $\mathcal{O}$ **do**
    mask all output positions
    for $t = 1, \dots, T$ **do**
        compute the teacher prediction at the masked positions, merged with CFG on image tasks
        commit the positions chosen by the teacher's schedule at this step
        record their sparse teacher predictions and the current step
        if $t \in \mathcal{F}$ **then**
            store a keyframe from the current state and the records accumulated since the previous one
        end if
    end for
    append the example to $\mathcal{D}$
end for

Algorithm 3 Stage 1: Off-Policy Trajectory Replay

Input: cache $\mathcal{D}$ (Algorithm 2), teacher p_ϕ, student $q_\theta \leftarrow p_\phi$, weights λ, training steps N_1
Output: few-step student q_θ, used to initialize Stage 2
for N_1 training steps **do**
    sample an example from $\mathcal{D}$ and two keyframes $x_t, x_{t'}$ with $t < t'$
    $\mathcal{L}_{\text{traj}} \leftarrow$ forward KL to the cached teacher prediction at the positions committed between them
    $\mathcal{L}_{\text{sc}} \leftarrow D_{\text{KL}}\big(\text{sg}[q_\theta(\cdot \mid x_{t'})] \,\|\, q_\theta(\cdot \mid x_t)\big)$
    $\mathcal{L}_{\text{path}} \leftarrow$ cross-entropy to the teacher's final tokens on a rebuilt intermediate state
    $\mathcal{L}_{\text{gt}} \leftarrow$ CE on freshly masked data
    update θ with $\mathcal{L}_1$ (Eq. 1)
end for

Algorithm 2 collects the teacher trajectories that Stage 1 replays. Algorithm 3 gives Stage 1, and Algorithm 1 in the main text gives Stage 2.

Sparse teacher predictions. Storing the full distribution at every cached position is too costly, so we keep the teacher's top-R log-probabilities and one residual probability for the rest of the vocabulary. Let $p_1, \ldots, p_R$ be the corresponding teacher probabilities and $p_{R+1} = 1 - \sum_k p_k$ the residual probability. Let $q_1, \ldots, q_R$ be the student's probabilities of the same tokens, with q_{R+1} defined in the same way. Both models are then compared on the same $R{+}1$ categories,

$$D_{\text{KL}}\big(p^{(R)} \,\|\, q^{(R)}\big) = \sum_{k=1}^{R} p_k \log \frac{p_k}{q_k} \;+\; p_{R+1} \log \frac{p_{R+1}}{q_{R+1}}. \tag{4}$$

Stage-1 terms. Let $x_t, x_{t'}$ with $t < t'$ be two keyframes, so x_t is the earlier one and has more masked positions. Let $\mathcal{U}$ be the positions the teacher commits between them, and $\mathcal{M}(x_{t'})$ the positions still masked at the later one. Then

$$\mathcal{L}_{\text{traj}} = \mathbb{E}\Big[\tfrac{1}{|\mathcal{U}|}\sum_{i\in\mathcal{U}} D_{\text{KL}}\big(p_\phi^{(R),i} \,\|\, q_\theta^{(R),i}\big)\Big], \qquad \mathcal{L}_{\text{sc}} = \mathbb{E}\Big[\tfrac{1}{|\mathcal{M}(x_{t'})|}\sum_{i\in\mathcal{M}(x_{t'})} D_{\text{KL}}\big(\text{sg}[q_\theta^i(\cdot \mid x_{t'})] \,\|\, q_\theta^i(\cdot \mid x_t)\big)\Big], \tag{5}$$

where the student predicts on x_t in both terms, and $p^{(R),i}$ is the sparse teacher prediction at position i. For the other two terms, let $\tilde{x}$ be an intermediate state rebuilt from the cached commit steps, with uncommitted positions $\mathcal{M}(\tilde{x})$. Let $\hat{y}_i$ be the token the teacher finally commits at position i. Let x be a randomly masked copy of a training output y, with masking rate π and output length n. Then

$$\mathcal{L}_{\text{path}} = \mathbb{E}\Big[\tfrac{1}{|\mathcal{M}(\tilde{x})|}\sum_{i\in\mathcal{M}(\tilde{x})} -\log q_\theta^i(\hat{y}_i \mid \tilde{x})\Big], \qquad \mathcal{L}_{\text{gt}} = \mathbb{E}\Big[\tfrac{1}{n}\sum_{i\in\mathcal{M}(x)} \tfrac{1}{\pi}\big(-\log q_\theta^i(y_i \mid x)\big)\Big]. \tag{6}$$

We weight by $1/\pi$ and divide by n instead of the number of masked positions. This makes $\mathcal{L}_{\text{gt}}$ an unbiased estimate of the full masked cross-entropy.

Image gradient and its surrogate loss. Distribution matching for continuous diffusion backpropagates through a reparameterized sample. The gradient of the divergence then becomes the difference between two scores (Yin et al., 2024b;a). Sampling discrete tokens is not differentiable, so this path does not exist. Following Di[M]O (Zhu et al., 2025a) for masked image generators, we specify the update directly instead of differentiating a divergence. The update $\text{sg}[r_\psi - p_\phi]$ in line 10 of Algorithm 1 is the categorical counterpart of the score difference. For a single categorical $q = \text{softmax}(z)$ and a target p, the forward KL has gradient $\nabla_z D_{\text{KL}}(p \,\|\, q) = q - p$. Our update replaces q with the student's own distribution at the scored state, estimated by r_ψ. We apply this update to z_θ without differentiating either network by minimizing

$$\mathcal{L}_{\text{match}}^{\text{img}} = \frac{1}{2|\mathcal{M}|}\big\|z_\theta - \text{sg}\big[z_\theta - g\big]\big\|^2, \qquad g = \text{sg}\big[r_\psi(\cdot \mid s) - p_\phi(\cdot \mid s)\big], \tag{7}$$

whose gradient is $g/|\mathcal{M}|$ by construction. This keeps the term on the same per-position scale as the other losses. We use the forward-KL direction because it encourages the student to cover the teacher's support. Both scores are computed on the re-masked output instead of the rollout state. On the rollout state, r_ψ would collapse onto q_θ, and the term would reduce to per-position knowledge distillation.

Entropy-matched guidance. Entropy increases with temperature, so we solve Eq. 2 for ν at each scored position by bisection on $[0.25, 8]$ with fourteen iterations. If the solution falls outside this range, we clip it to the nearest endpoint.

Preference term. $\mathcal{L}_{\text{pref}}$ follows T3D (Zhang et al., 2026), which applies direct discriminative optimization (Zheng et al., 2025) to few-step text dLLMs. It trains the student to prefer a completion y^+ over a sample y^- (line 19 of Algorithm 1). The preferred completion y^+ is the teacher's completion of the state. The dispreferred y^- is sampled from a reference q_{ref}, which is copied from the student every τ updates. Let $\Delta_\theta(y) = \frac{1}{|\mathcal{M}|}\sum_{i\in\mathcal{M}} \log\big(q_\theta^i(y_i \mid s)/q_{\text{ref}}^i(y_i \mid s)\big)$ be the mean log-ratio between the student and the reference. Then

$$\mathcal{L}_{\text{pref}} = -\log\sigma\big(\beta\,\Delta_\theta(y^+)\big) + \alpha\,\text{softplus}\big(\beta\,\Delta_\theta(y^-)\big), \tag{8}$$

where σ is the logistic function and α, β are scalar weights. For images, the term is computed on the rollout state x_t and its masked positions instead of s and $\mathcal{M}$.

Inference. The student is trained on CFG targets, so it generates images in a few decoding steps without an unconditional forward pass. For understanding, it decodes block by block and reuses the KV cache exactly. This relies on the block-causal mask. The teacher uses fully bidirectional attention and cannot reuse its cache.

C Experimental Details

C.1 Training Data and Teacher Trajectories

Training data. The training mixture has 51K examples. T2I data (11K) come from BLIP3o-60k (Chen et al., 2025a) and the Lumina-DiMOO training data. I2I data (22K) come from UltraEdit (Zhao et al., 2024), Subjects200K (Tan et al., 2025), Graph-200K (Li et al., 2025c), UniWorld-V1 (Lin et al., 2025) and the Lumina-DiMOO training data. MMU data (13K) come from MAmmoTH-VL (Guo et al., 2025), MultiUI (Liu et al., 2025d) and the Lumina-DiMOO training data. Inpainting pairs (5K) are built from the T2I images by masking a centered region and serve only as extra training data. Each mini-batch contains one task, sampled at the rates in Table 6.

Teacher trajectories. We run Algorithm 2 once over the mixture, and every Stage-1 run reuses the resulting cache. Image trajectories use T=64 steps with classifier-free guidance. The guidance strength w_{task} is 4 on the prompt for T2I. For editing, it is 2.5 on the instruction and 4 on the source image. Understanding trajectories use T=512 steps without guidance. The full-step teacher in Section 4.1 is evaluated at 128 image steps.

C.2 Optimization and Cost

Table 6 gives the full training configuration. The two stages share the data mixture, the batch size and the ground-truth anchor. They differ in the loss terms and the optimization settings.

Table 7 reports the one-time training cost, and Table 8 reports the cost per output at the deployed budget. The wall-clock speedup differs from the NFE reduction. Understanding speeds up by 21.2×, more than its 8× NFE reduction, because the block-causal mask allows exact KV cache reuse (Appendix B). Each block reuses the keys and values of the prompt and all earlier finished blocks, and recomputes only the current block and the one just finished. Without the cache, the same student is only 6.2× faster. T2I speeds up by 18.2×, less than its 32× NFE reduction. Peak memory is the same for both models, between 16 and 21 GB.

C.3 Baselines

All baselines are distilled from the same Lumina-DiMOO teacher with the same initialization, training mixture and number of optimizer updates. They are evaluated with the same sampler at the same decoding budgets, and each keeps its own published objective. Every method is trained once and evaluated at all budgets from that single checkpoint, with no budget-specific retraining or tuning. The only change is modality coverage. Di[M]O was designed for masked image generators, and CDLM and T3D for text dLLMs. We apply each of them to the other modality as well.

Table 6: **Training Configuration.** The collision penalty applies only to the text branch. The auxiliary network r_ψ and the student reference q_{ref} are used only in Stage 2 and are discarded after training.

	Stage 1 (off-policy)	Stage 2 (on-policy)
Optimizer	AdamW (β_1=0.9, β_2=0.95)	
Learning rate (cosine)	$2\times10^{-5}\rightarrow2\times10^{-6}$	$1\times10^{-5}\rightarrow1\times10^{-6}$
Weight decay	0.1	0
Warmup updates	50	25
Global batch size	256	
Attention mask (text / image)	block-causal / bidirectional	
Task sampling prob. (T2I / MMU / I2I / Inpaint)	0.50 / 0.25 / 0.125 / 0.125	
Image resolution (T2I / I2I, inpaint)	1024×1024 / 512×512	
Text block size / output length	32 / 512	
Decoding steps per text block	4	
Trajectory decoding steps (image / text)	64 / 512	
Loss weights, Stage 1	λ_{traj} 1.0, λ_{sc} 0.5, λ_{path} 0.2, λ_{gt} 0.1	
Loss weights, Stage 2	λ_{match} 0.5, λ_{col} 8.0, λ_{pref} 0.2, λ_{gt} 0.1	
Keyframes per example (image / text)	8 / 64	N/A
Retained probabilities R (image / text)	512 / 2048	N/A
Re-mask fraction range	N/A	$[0.2, 0.9]$
Entropy fraction ρ	N/A	0.7
Reference refresh τ (updates)	N/A	20
Preference α / β	N/A	0.5 / 0.1

Table 7: **Training Cost.** One-time cost on 32 A100 GPUs. Trajectory collection runs once over the training mixture, and every Stage-1 run reuses it.

Training	Wall-clock	GPU-hours	Models in memory
Teacher trajectory collection	50.5 h	1,616	p_ϕ
Stage 1 (off-policy replay)	29.0 h	928	q_θ
Stage 2 (on-policy refinement)	31.3 h	1,002	q_θ, p_ϕ, r_ψ, q_{ref}
Total	110.8 h	3,546	

C.4 Benchmarks

T2I evaluation covers GenEval (Ghosh et al., 2023), DPG-Bench (Hu et al., 2024), UniGen-Bench (Wang et al., 2025b;a), OneIG (Chang et al., 2025) with English prompts and TIIF-Bench (Wei et al., 2025b). I2I is evaluated on the I2I benchmark and Graph-200K (Li et al., 2025c). For MMU, the discriminative benchmarks are POPE (Li et al., 2023), MME-P (Fu et al., 2023), MM-Bench (Liu et al., 2024b), SEED (Li et al., 2024) and MMMU (Yue et al., 2024). The open-ended benchmarks are MM-Vet (Yu et al., 2024), Vibe-Eval (Padlewski et al., 2024), LLaVA-Bench (Liu et al., 2023a), COCO captioning (Chen et al., 2015) and DeCapBench (Ye et al., 2025b). Each benchmark uses its official metric, except COCO captioning, for which we define our own protocol in Appendix C.5.

C.5 Open-Ended Judging Protocols

All five open-ended benchmarks are scored by the same GPT-5.5 judge. Four of them keep their official grading prompt and aggregation. For COCO captioning, we define our own protocol.

- **MM-Vet** (Yu et al., 2024). We use the official soft-grading prompt. The judge sees the question, the ground-truth answer and the prediction, and gives a correctness score in $\{0.0, 0.1, \ldots, 1.0\}$. We report the mean $\times100$.

Table 8: **Inference Cost per Output.** Measured on one A100-80GB at batch size 1 in bfloat16. We time only the token-space sampler and report the mean of 10 runs after 2 warmup runs. Parentheses give the student's speedup over the teacher. TPS is tokens per second.

	NFE ↓		Latency (s) ↓		TPS ↑	
Inference (per output)	Teacher	Student	Teacher	Student	Teacher	Student
T2I	128×2=256	8 (32×)	106.4	5.84 (18.2×)	38.5	701.4
I2I	128×3=384	8 (48×)	79.6	2.28 (34.9×)	12.9	449.7
MMU	512×1=512	64 (8×)	80.7	3.81 (21.2×)	6.3	134.5
w/o KV cache	512×1=512	64 (8×)	80.7	13.12 (6.2×)	6.3	39.0

- **Vibe-Eval** (Padlewski et al., 2024). We use the official text-only grading prompt. The judge rates the output against the reference on a 1 to 5 scale, and we rescale the score s to 0 to 100 by $25(s-1)$.
- **LLaVA-Bench** (Liu et al., 2023a). We use the standard pairwise protocol. The judge sees the image caption instead of the image and scores both the model answer and the reference GPT-4 answer from 1 to 10. We report the relative score, which is the model's mean score divided by the reference's mean score, times 100.
- **COCO captioning** (Chen et al., 2015). This protocol is ours. The judge sees the image and rates each caption on accuracy, completeness and coherence from 1 to 5. It is told to penalize hallucinated objects and not to reward length. The five reference captions are given as extra context, not as ground truth. Scores are rescaled to 0 to 100 by $25(s-1)$. We use a fixed subset of 1000 validation images.
- **DeCapBench** (Ye et al., 2025b). We use the official DCSCORE pipeline. It splits a caption into atomic facts, matches them against the oracle facts and checks each one against the image. The score is the mean of two F1 scores, one over all atomic facts and one over descriptive facts only, $\times 100$.

The discriminative benchmarks use no judge. For yes/no benchmarks, we search the output for a standalone "yes" or "no". For multiple-choice benchmarks, we match a standalone option letter and fall back to the full option string. This parsing makes the scores insensitive to the rest of the output.

D Additional Results

This section extends Section 4 with results at smaller decoding budgets, results in each category of the T2I benchmarks and more qualitative samples.

D.1 Results at Smaller Decoding Budgets

Tables 9 to 12 report T2I, I2I and the two MMU groups at the smaller decoding budgets that Section 4 omits. GenEval at all budgets is in Table 13. On T2I and MMU, the results follow the same pattern as at the larger budgets. On T2I, Omni-Diffusion-Distill leads every benchmark at every budget, and its margin generally grows as the budget shrinks. On discriminative MMU, the scores stay flat down to 16 steps, and Omni-Diffusion-Distill is the best distilled method on all five benchmarks. On open-ended MMU, the scores fall for every method. Omni-Diffusion-Distill keeps the largest margin on COCO captioning, while T3D is higher on LLaVA-Bench at 32 and 16 steps. On I2I, Omni-Diffusion-Distill leads Add at both budgets and Graph-200K at 4 steps. On the I2I benchmark Overall, T3D is higher at both budgets and Di[M]O at 4 steps, by at most 0.07. T3D is also higher on Graph-200K at 2 steps.

D.2 Results in Each Category of the T2I Benchmarks

Tables 13, 14, 15, 16 and 17 report GenEval, DPG-Bench, UniGenBench, OneIG and TIIF-Bench in each category, following the official protocol of each benchmark.

Table 9: **Quantitative Results on T2I at Smaller Decoding Budgets.** The format follows Table 1. GenEval at all budgets is in Table 13.

Method	NFE ↓	DPG-Bench ↑	UniGenBench ↑	OneIG ↑	TIIF-Bench ↑
Teacher (full-step)	128×2	83.8	0.722	0.445	0.781
4 image decoding steps					
T3D	4×1	70.6	0.630	0.283	0.641
Di[M]O	4×1	78.0	0.689	0.296	0.716
CDLM	4×1	51.3	0.450	0.212	0.547
Omni-Diffusion-Distill	4×1	**83.1**	**0.724**	**0.368**	**0.759**
Teacher (few-step)	4×2	63.3	0.559	0.253	0.598
2 image decoding steps					
T3D	2×1	55.9	0.524	0.214	0.591
Di[M]O	2×1	55.7	0.564	0.225	0.603
CDLM	2×1	13.5	0.119	0.099	0.179
Omni-Diffusion-Distill	2×1	**73.8**	**0.677**	**0.277**	**0.654**
Teacher (few-step)	2×2	37.0	0.344	0.174	0.484

Table 10: **Quantitative Results on I2I at Smaller Decoding Budgets.** The format follows Table 2.

Method	NFE ↓	I2I Benchmark					Graph-200K
		Add ↑	Compose ↑	Adjust ↑	Style ↑	Overall ↑	DINOv2 ↑
Teacher (full-step)	128×3	3.62	1.94	2.29	3.89	2.66	0.799
4 image decoding steps							
T3D	4×1	2.20	1.80	**2.49**	3.74	**2.30**	0.773
Di[M]O	4×1	2.12	1.76	2.16	**3.78**	2.25	0.768
CDLM	4×1	1.33	1.07	1.39	2.92	1.53	0.513
Omni-Diffusion-Distill	4×1	**2.91**	**1.84**	2.42	**3.78**	2.24	**0.783**
Teacher (few-step)	4×3	3.01	1.65	2.28	3.84	2.46	0.758
2 image decoding steps							
T3D	2×1	2.23	**1.87**	2.32	**3.63**	**2.26**	**0.732**
Di[M]O	2×1	2.27	1.42	2.10	3.47	2.13	0.681
CDLM	2×1	1.00	1.00	1.00	1.41	1.15	0.111
Omni-Diffusion-Distill	2×1	**2.86**	1.58	**2.51**	3.50	2.19	0.710
Teacher (few-step)	2×3	2.40	1.51	2.11	3.46	2.19	0.668

At 8 decoding steps, Omni-Diffusion-Distill leads every overall score. Its largest margins are in categories where several parts of a prompt must be satisfied at once, namely Two Obj. and Position on GenEval, Other on DPG-Bench, Relation and Compound on UniGenBench, and Real-world on TIIF-Bench. It also holds up on long prompts, with 0.764 on TIIF-Bench Long and 0.772 on Short, while the strongest baseline drops from 0.719 to 0.691. In several categories it matches or exceeds the full-step teacher, including DPG-Bench Other (86.8 against 85.2), UniGenBench Relation (0.828 against 0.792) and Compound (0.778 against 0.740). The remaining gap lies mainly in text rendering, where OneIG Text is 0.259 against the teacher's 0.542, and in the holistic DPG-Bench Global category.

D.3 Additional Qualitative Samples

Figure 8 shows more T2I samples in the same format as Figure 4, and Figure 9 shows controllable generation samples.

E Analysis and Ablations

This section checks whether the collision penalty and entropy-matched guidance work as intended, and which design choices matter for the reported results.

Table 11: **Quantitative Results on Discriminative MMU at Smaller Decoding Budgets.** The format follows Table 3.

Method	NFE ↓	POPE ↑	MME-P ↑	MMBench ↑	SEED ↑	MMMU ↑
Teacher (full-step)	512	86.8	1552	91.2	85.3	58.3
32 decoding steps						
T3D	32	82.8	1387	60.1	79.0	41.4
Di[M]O	32	86.3	1478	57.6	76.4	36.0
CDLM	32	79.9	1318	78.5	79.3	50.3
Omni-Diffusion-Distill	32	**87.6**	**1535**	**89.2**	**84.5**	**56.2**
Teacher (few-step)	32	86.5	1571	91.0	84.9	58.8
16 decoding steps						
T3D	16	83.5	1419	60.9	77.2	41.6
Di[M]O	16	86.0	1515	61.9	77.7	43.3
CDLM	16	85.3	1480	71.8	73.2	49.8
Omni-Diffusion-Distill	16	**87.3**	**1544**	**89.2**	**84.8**	**56.0**
Teacher (few-step)	16	86.5	1551	90.6	85.2	58.4

Table 12: **Quantitative Results on Open-Ended MMU at Smaller Decoding Budgets.** The format follows Table 4.

Method	NFE ↓	MM-Vet ↑	Vibe-Eval ↑	LLaVA-Bench ↑	COCO ↑	DeCapBench ↑
Teacher (full-step)	512	21.3	12.36	25.1	67.2	44.5
32 decoding steps						
T3D	32	14.9	3.90	**17.2**	13.5	8.9
Di[M]O	32	13.2	3.53	13.2	24.4	10.5
CDLM	32	15.6	3.90	13.3	31.4	10.6
Omni-Diffusion-Distill	32	**16.8**	**6.41**	13.7	**40.0**	**13.2**
Teacher (few-step)	32	15.8	3.62	12.2	17.3	10.3
16 decoding steps						
T3D	16	11.6	3.25	**13.8**	8.9	7.5
Di[M]O	16	11.3	2.70	12.2	15.1	8.5
CDLM	16	14.1	3.07	12.2	18.1	5.9
Omni-Diffusion-Distill	16	**15.0**	**4.09**	13.1	**26.0**	**10.4**
Teacher (few-step)	16	13.4	2.70	12.0	12.4	8.9

E.1 Why the Collision Penalty and Entropy-Matched Guidance Help

The collision penalty and entropy-matched guidance in Section 3.4 address the two challenges in Section 1. These are repetition in parallel text decoding and entropy collapse caused by fitting the sharpened CFG target in image distillation. Table 18 measures both directly.

Text degeneration metrics. We compute degeneration metrics on the outputs of the five open-ended understanding benchmarks and average them across benchmarks. For a generated token sequence $y_1, \ldots, y_n$, we report four metrics.

- Rep-1 is the fraction of adjacent token pairs with $y_i = y_{i-1}$. It measures local token repetition.
- Rep-4 is $1 - |\{\text{unique 4-grams}\}|/|\{\text{4-grams}\}|$. It measures repeated phrases instead of repeated tokens.
- Loop@5 is the fraction of outputs that contain at least five identical tokens in a row. It captures severe repetition loops.
- Distinct-1 is the ratio of unique tokens to all tokens in an output. Higher values mean more diverse outputs.

Output lengths differ across models, so we compute Rep-4 and Distinct-1 on the first 64 tokens of each output to compare them at equal length.

Table 13: **Results in Each Category of GenEval.** Overall is the mean of the six categories. ImageReward (Xu et al., 2023) and Aesthetic are image quality metrics computed on the same generated images.

Method	NFE ↓	Single Obj.	Two Obj.	Counting	Colors	Position	Attribute	Overall ↑	Image Reward ↑	Aesthetic ↑
Teacher (full-step)	128×2	0.984	0.934	0.822	0.896	0.825	0.743	0.867	1.33	3.65
8 image decoding steps										
T3D	8×1	0.947	0.846	0.700	0.814	0.770	0.573	0.775	0.97	3.55
Di[M]O	8×1	0.972	0.848	**0.747**	0.840	0.772	**0.677**	0.810	1.12	3.61
CDLM	8×1	0.806	0.518	0.503	0.585	0.403	0.310	0.521	0.29	3.33
Omni-Diffusion-Distill	8×1	**0.984**	**0.891**	0.713	**0.864**	**0.835**	**0.677**	**0.828**	**1.25**	**3.71**
Teacher (few-step)	8×2	0.978	0.811	0.672	0.832	0.743	0.670	0.784	0.98	3.61
4 image decoding steps										
T3D	4×1	0.966	0.770	**0.709**	0.809	0.700	0.562	0.753	0.85	3.50
Di[M]O	4×1	0.944	0.747	0.647	0.795	0.665	0.555	0.726	0.82	3.52
CDLM	4×1	0.622	0.164	0.228	0.465	0.128	0.128	0.289	-0.73	3.04
Omni-Diffusion-Distill	4×1	**0.969**	**0.851**	0.641	**0.878**	**0.807**	**0.640**	**0.798**	**1.15**	**3.69**
Teacher (few-step)	4×2	0.794	0.376	0.356	0.614	0.420	0.270	0.472	-0.02	3.47
2 image decoding steps										
T3D	2×1	0.803	0.333	0.378	0.596	0.347	0.207	0.444	-0.11	3.12
Di[M]O	2×1	0.694	0.232	0.281	0.503	0.217	0.125	0.342	-0.58	2.96
CDLM	2×1	0.100	0.000	0.003	0.053	0.000	0.000	0.026	-2.11	2.86
Omni-Diffusion-Distill	2×1	**0.884**	**0.505**	**0.419**	**0.710**	**0.485**	**0.318**	**0.553**	**0.25**	**3.38**
Teacher (few-step)	2×2	0.466	0.051	0.084	0.285	0.077	0.020	0.164	-1.27	3.22

Table 14: **Results in Each Category of DPG-Bench.** Overall is the official DPG-Bench score computed over all prompts.

Method	NFE ↓	Global	Entity	Attribute	Relation	Other	Overall ↑
Teacher (full-step)	128×2	81.5	90.5	88.2	93.0	85.2	83.8
8 image decoding steps							
T3D	8×1	69.0	79.2	79.7	85.6	73.6	69.1
Di[M]O	8×1	**84.2**	87.6	86.5	92.0	76.8	80.8
CDLM	8×1	71.4	77.0	77.6	86.2	64.8	66.2
Omni-Diffusion-Distill	8×1	77.8	**89.8**	**87.7**	**92.9**	**86.8**	**83.0**
Teacher (few-step)	8×2	77.8	85.9	84.7	91.8	75.6	78.6
4 image decoding steps							
T3D	4×1	72.6	80.0	80.9	88.0	75.2	70.6
Di[M]O	4×1	79.6	85.7	85.2	91.0	67.6	78.0
CDLM	4×1	64.7	62.3	67.2	80.5	40.4	51.3
Omni-Diffusion-Distill	4×1	**81.8**	**89.7**	**88.1**	**92.6**	**84.4**	**83.1**
Teacher (few-step)	4×2	71.7	72.6	77.5	86.5	56.0	63.3
2 image decoding steps							
T3D	2×1	68.1	67.7	72.8	82.4	52.8	55.9
Di[M]O	2×1	74.5	67.8	72.2	83.2	45.6	55.7
CDLM	2×1	52.3	20.0	40.0	44.3	7.6	13.5
Omni-Diffusion-Distill	2×1	**79.0**	**82.1**	**82.1**	**89.7**	**71.6**	**73.8**
Teacher (few-step)	2×2	64.7	49.7	64.0	76.8	33.2	37.0

Image predictive entropy. On the image side, classifier-free guidance produces a sharpened target, and we measure its effect on the student directly. At the final decoding step, we compute the entropy of the predicted distribution at every masked image position and average it over positions and samples to get H_{final}. For the teacher, we measure the raw CFG distribution $\mathrm{softmax}(\ell^{\text{cfg}})$ before Eq. 2. For the student, we measure its conditional prediction, since the distilled student reproduces the guided distribution from a conditional forward pass alone.

Table 15: **Results in Each Category of UniGenBench.** Overall is the mean of the ten categories. World Know. is world knowledge, and Logic is logical reasoning.

Method	NFE	Style	World Know.	Attribute	Action	Relation	Logic	Grammar	Compound	Layout	Text	Overall
Teacher (full-step)	128×2	0.936	0.870	0.876	0.735	0.792	0.432	0.701	0.740	0.873	0.267	0.722
8 image decoding steps												
T3D	8×1	0.850	0.752	0.788	0.656	0.682	0.366	0.682	0.651	0.808	**0.163**	0.640
Di[M]O	8×1	**0.943**	0.851	0.865	0.740	0.768	0.435	**0.713**	0.733	0.859	0.095	0.700
CDLM	8×1	0.798	0.644	0.664	0.554	0.561	0.293	0.595	0.543	0.674	0.035	0.536
Omni-Diffusion-Distill	8×1	0.901	**0.869**	**0.873**	**0.752**	**0.828**	**0.453**	0.694	**0.778**	**0.875**	0.082	**0.710**
Teacher (few-step)	8×2	0.917	0.810	0.824	0.709	0.753	0.424	0.677	0.737	0.846	0.067	0.676
4 image decoding steps												
T3D	4×1	0.887	0.737	0.783	0.654	0.674	0.361	0.676	0.641	0.794	**0.096**	0.630
Di[M]O	4×1	0.923	0.834	0.849	0.727	0.776	0.438	**0.716**	0.745	0.847	0.034	0.689
CDLM	4×1	0.711	0.478	0.563	0.434	0.498	0.255	0.511	0.457	0.586	0.004	0.450
Omni-Diffusion-Distill	4×1	**0.940**	**0.881**	**0.889**	**0.772**	**0.833**	**0.482**	0.706	**0.798**	**0.870**	0.064	**0.724**
Teacher (few-step)	4×2	0.851	0.628	0.690	0.544	0.632	0.319	0.579	0.622	0.709	0.013	0.559
2 image decoding steps												
T3D	2×1	0.818	0.585	0.657	0.502	0.578	0.285	0.584	0.545	0.674	0.015	0.524
Di[M]O	2×1	0.790	0.646	0.669	0.557	0.636	0.334	0.658	0.602	0.736	0.008	0.564
CDLM	2×1	0.351	0.054	0.171	0.061	0.115	0.042	0.184	0.095	0.115	0.000	0.119
Omni-Diffusion-Distill	2×1	**0.872**	**0.793**	**0.836**	**0.725**	**0.792**	**0.446**	**0.704**	**0.787**	**0.787**	**0.024**	**0.677**
Teacher (few-step)	2×2	0.648	0.285	0.471	0.260	0.374	0.148	0.430	0.348	0.474	0.003	0.344

Table 16: **Results in Each Category of OneIG.** Overall is the mean of the five categories.

Method	NFE ↓	Alignment	Text	Reasoning	Style	Diversity	Overall ↑
Teacher (full-step)	128×2	0.817	0.542	0.281	0.400	0.185	0.445
8 image decoding steps							
T3D	8×1	0.667	0.122	0.155	0.365	0.218	0.306
Di[M]O	8×1	0.780	0.095	0.164	0.375	0.223	0.327
CDLM	8×1	0.577	0.029	0.148	0.318	**0.330**	0.280
Omni-Diffusion-Distill	8×1	**0.801**	**0.259**	**0.245**	**0.393**	0.206	**0.381**
Teacher (few-step)	8×2	0.725	0.200	0.210	0.375	0.275	0.357
4 image decoding steps							
T3D	4×1	0.652	0.066	0.150	0.335	0.210	0.283
Di[M]O	4×1	0.744	0.021	0.163	0.345	0.209	0.296
CDLM	4×1	0.420	0.003	0.122	0.273	**0.243**	0.212
Omni-Diffusion-Distill	4×1	**0.803**	**0.221**	**0.229**	**0.385**	0.204	**0.368**
Teacher (few-step)	4×2	0.531	0.010	0.147	0.317	0.259	0.253
2 image decoding steps							
T3D	2×1	0.468	0.007	0.131	**0.300**	0.166	0.214
Di[M]O	2×1	0.519	0.002	0.142	0.294	0.166	0.225
CDLM	2×1	0.123	0.000	0.099	0.169	0.104	0.099
Omni-Diffusion-Distill	2×1	**0.680**	**0.028**	**0.168**	0.293	**0.213**	**0.277**
Teacher (few-step)	2×2	0.310	0.000	0.122	0.271	0.167	0.174

Results. On the text side, aggressive step reduction greatly increases degeneration. With the same teacher weights, Loop@5 rises from 0.6% with full-step decoding to 74.4% at the student's decoding budget. Omni-Diffusion-Distill reduces it to 7.7%. Without the collision penalty, it rises to 21.6%, with higher Rep-1 and Rep-4 and lower Distinct-1. The collision penalty therefore directly reduces the repetition caused by parallel text decoding.

On the image side, without entropy-matched guidance, the student's final entropy falls to 1.55. This is even below the 2.28 of the raw CFG target of the few-step teacher at the same budget. Entropy-matched guidance raises it to 3.32, above the raw CFG target at both budgets.

Table 17: **Results in Each Category of TIIF-Bench.** Short and Long are the overall scores on the two prompt lengths, and Overall is their mean. The three category columns are scored on short prompts.

Method	NFE ↓	Short	Long	Basic	Advanced	Real-world	Overall ↑
Teacher (full-step)	128×2	0.808	0.755	0.869	0.813	0.690	0.781
8 image decoding steps							
T3D	8×1	0.670	0.655	0.828	0.719	0.493	0.663
Di[M]O	8×1	0.719	0.691	**0.852**	0.762	0.552	0.705
CDLM	8×1	0.616	0.582	0.795	0.688	0.470	0.599
Omni-Diffusion-Distill	8×1	**0.772**	**0.764**	0.847	**0.806**	**0.657**	**0.768**
Teacher (few-step)	8×2	0.723	0.676	0.853	0.771	0.545	0.699
4 image decoding steps							
T3D	4×1	0.640	0.641	0.819	0.707	0.437	0.641
Di[M]O	4×1	0.732	0.700	**0.857**	0.794	0.545	0.716
CDLM	4×1	0.579	0.515	0.812	0.659	0.358	0.547
Omni-Diffusion-Distill	4×1	**0.784**	**0.735**	0.856	**0.809**	**0.653**	**0.759**
Teacher (few-step)	4×2	0.618	0.578	0.757	0.704	0.444	0.598
2 image decoding steps							
T3D	2×1	0.573	0.609	0.790	0.649	0.321	0.591
Di[M]O	2×1	0.623	0.583	**0.847**	0.717	0.369	0.603
CDLM	2×1	0.160	0.199	0.244	0.181	0.056	0.179
Omni-Diffusion-Distill	2×1	**0.654**	**0.654**	0.782	**0.731**	**0.560**	**0.654**
Teacher (few-step)	2×2	0.509	0.459	0.643	0.629	0.317	0.484

Table 18: **Analysis of Text Repetition and Image Entropy.** Left: degeneration metrics on the open-ended understanding benchmarks. Right: predictive entropy of image tokens at the final decoding step. Teacher rows use the raw CFG target, and student rows use the student's conditional prediction.

Model	**Text degeneration**				**Image entropy**
	Rep-1 ↓	Rep-4 ↓	Loop@5 ↓	Distinct-1 ↑	H_{final} (nats)
Teacher (full-step)	0.020	0.043	0.006	0.704	2.13
Teacher (few-step)	0.473	0.459	0.744	0.313	2.28
Omni-Diffusion-Distill	**0.091**	**0.020**	**0.077**	**0.688**	3.32
w/o collision penalty	0.178	0.122	0.216	0.592	–
w/o entropy-matched guidance	–	–	–	–	1.55

Eq. 2 solves for one temperature at each position and step. A simpler alternative is one global temperature for the whole trajectory. Figure 10 compares entropy-matched guidance with fixed-temperature scaling. Entropy-matched guidance tracks the decreasing conditional entropy at the fraction $\rho = 0.7$ throughout decoding. A fixed temperature cannot track this decrease. With either fixed temperature, the entropy falls below the entropy-matched target early in decoding and exceeds it near the end. Table 21 shows that a fixed temperature also lowers the benchmark scores.

Both measurements are consistent with the ablations in Tables 20 and 21. The collision penalty brings its largest gain on COCO captioning, and entropy-matched guidance brings its largest gain on the I2I benchmark. This matches what each component acts on. The collision penalty acts on the probability that two scored positions in a block commit the same token, and entropy-matched guidance acts on the entropy of the teacher prediction.

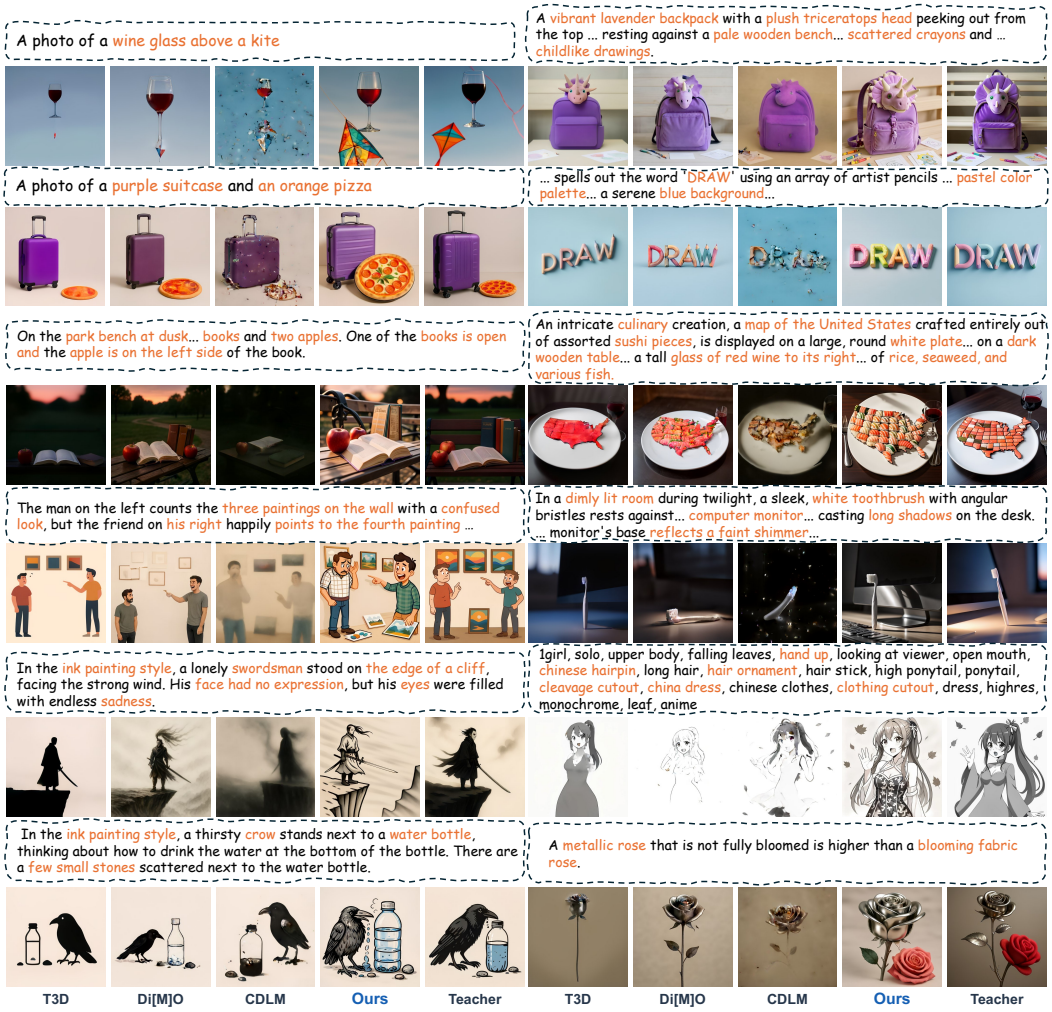


Figure 8: **Additional qualitative comparison on T2I.** All methods use the same prompts and seeds. Distilled models use 8 decoding steps, and the teacher uses 128.

E.2 Ablations

Table 19 tests whether both training stages are needed, Table 20 varies the understanding objective, and Table 21 varies the entropy fraction ρ and compares entropy-matched guidance with a fixed temperature.

Table 19: **Effect of the Two Training Stages.** Rows compare Stage 1 only, Stage 2 only without the Stage 1 cold start, and the full two-stage training.

Training	Text-to-image					Image-to-image		Multimodal understanding		
	GenEval ↑	DPG-Bench ↑	OneIG ↑	TIIF-Bench ↑	UniGenBench ↑	I2I Benchmark ↑	Graph-200K ↑	MM-Vet ↑	Vibe-Eval ↑	COCO ↑
Stage 1 only	0.740	68.2	0.282	0.585	0.585	2.10	0.765	17.0	7.34	36.1
Stage 2 only	0.771	81.8	0.333	0.719	0.684	2.15	0.769	19.4	7.62	56.5
Ours	**0.828**	**83.0**	**0.381**	**0.768**	**0.710**	**2.31**	**0.789**	**20.0**	**8.18**	**57.2**

Both stages. Both stages are needed. Stage 1 alone scores below the few-step teacher on every image benchmark. Its role is to give Stage 2 a good starting point, not to serve as the final model. Using Stage 1 as a cold start for Stage 2 improves every benchmark. The largest gain is 14.4% on OneIG, followed by 7.4% on GenEval, 7.4% on the I2I benchmark and 7.3% on Vibe-Eval. The

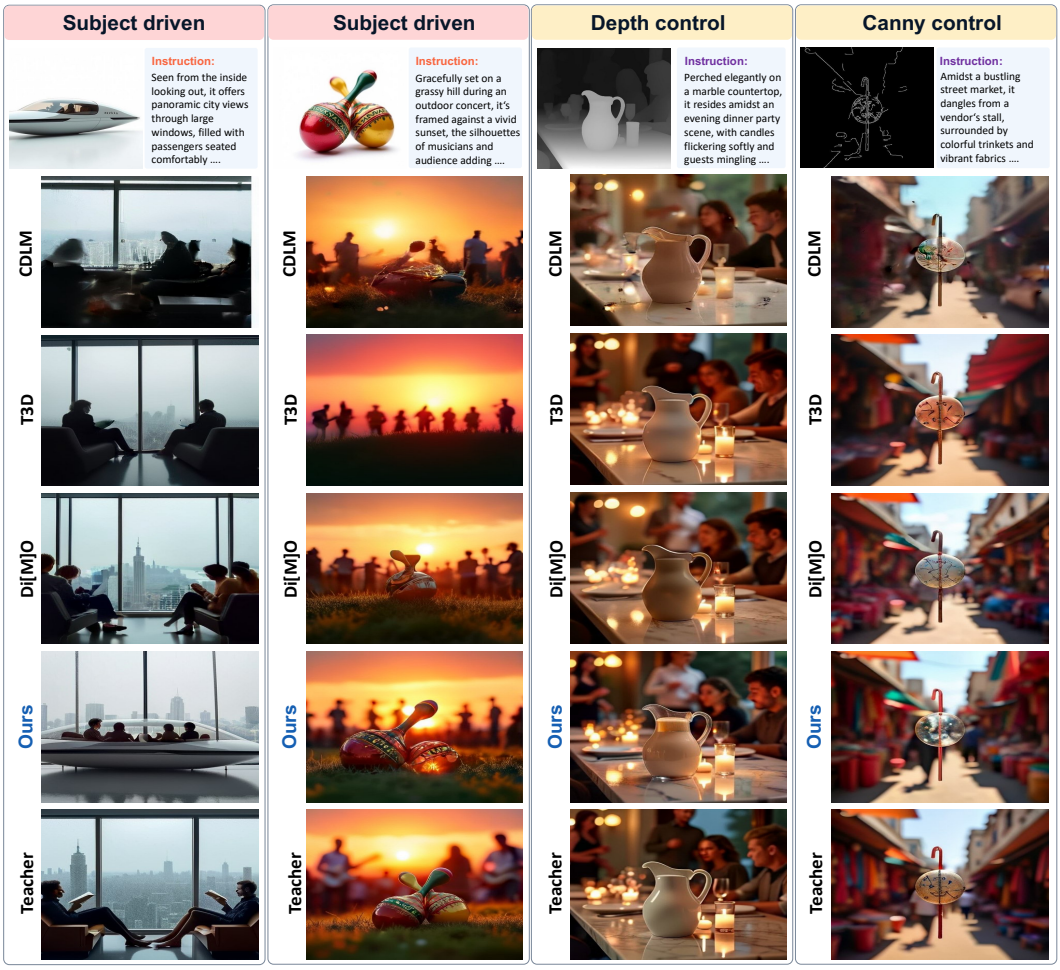


Figure 9: **Additional qualitative comparison on controllable generation.** All methods use the same conditioning inputs and prompts. Distilled models use 8 decoding steps, and the teacher uses 128.

Table 20: **Ablation on the MMU Objective.** Rows vary the scored state, the form of the teacher prediction and the repetition term.

Scored state	Teacher prediction	Repetition term	MM-Vet ↑	Vibe-Eval ↑	COCO ↑	DeCapBench ↑
Sampler's own masked positions	one-step marginal	none	18.8	6.78	47.3	14.9
	sequential conditional	none	18.2	8.09	49.8	13.5
		adjacent repeat	18.9	7.62	49.3	14.4
		pairwise collision	**20.0**	**8.18**	**57.2**	19.6
Re-masked committed tokens	sequential conditional	none	16.9	7.90	48.5	13.7
		adjacent repeat	16.8	7.53	49.5	14.6
		pairwise collision	19.4	8.18	53.4	**21.2**

gains are smaller on MM-Vet, from 19.4 to 20.0, and on COCO captioning, from 56.5 to 57.2. Without Stage 2, the student scores 14.8 points lower on DPG-Bench and 21.1 lower on COCO captioning.

Understanding objective. The scored state decides where the student is trained. With the sampler's own masked positions, which is the state used in Section 3.4.2, we score the positions that the current block has not yet committed at a random step of the student's rollout. With re-masked committed tokens, we instead take positions the student has already committed and mask them

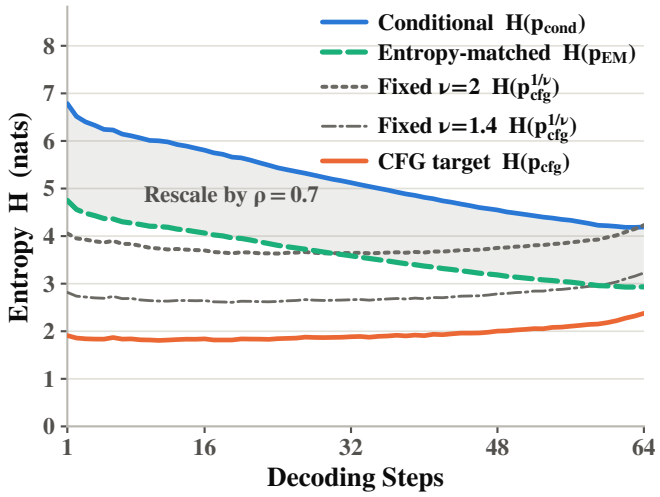


Figure 10: **Entropy trajectories under adaptive and fixed-temperature scaling.** Entropy-matched guidance targets 0.7 times the conditional entropy along the 64-step teacher trajectory. The two fixed temperatures produce lower entropy than this target early in decoding and higher entropy near the end.

again. With the pairwise collision penalty, the sampler's own masked positions give higher scores on MM-Vet and COCO captioning, the same score on Vibe-Eval and a lower score on DeCapBench.

The teacher prediction and the repetition term have larger effects. The one-step marginal queries the teacher once on the state, so the prediction at each position ignores the tokens committed at the other positions. The sequential conditional lets the teacher commit these positions one at a time, so each prediction sees the tokens the teacher has already committed. This change improves Vibe-Eval from 6.78 to 8.09 and COCO captioning from 47.3 to 49.8, while MM-Vet and DeCapBench drop slightly. It does not remove repetition by itself, as Section 3.4.2 predicts, because a per-position prediction has no explicit term for two positions in a block committing the same token. The largest improvement comes from the pairwise collision penalty, which acts on every pair of scored positions in a block. It raises COCO captioning to 57.2 and DeCapBench to 19.6. An adjacent-repeat penalty acts only on neighboring positions and thus directly optimizes Rep-1, but it brings almost no improvement.

Table 21: **Ablation on Entropy-Matched Guidance.** The entropy fraction ρ is relative to the entropy of the conditional distribution at each scored position. Off disables entropy-matched guidance and uses the raw CFG target. Fixed ν divides the guided logits by one temperature at all positions and steps.

Setting	GenEval ↑	DPG-Bench ↑	UniGenBench ↑	OneIG ↑	TIIF-Bench ↑	ImageReward ↑	Aesthetic ↑	I2I Benchmark ↑
Off	0.798	81.5	0.690	0.343	0.723	1.23	**3.73**	2.04
$\rho = 0.5$	0.815	79.0	0.704	0.377	0.745	1.19	3.65	2.20
$\rho = 0.7$ (Ours)	**0.828**	**83.0**	**0.710**	**0.381**	**0.768**	**1.25**	3.71	**2.31**
$\rho = 0.9$	0.176	67.5	0.562	0.260	0.549	−0.89	3.33	2.00
Fixed $\nu = 1.4$	0.475	78.6	0.670	0.304	0.707	0.72	3.52	2.03
Fixed $\nu = 2.0$	0.015	26.2	0.185	0.137	0.298	−2.11	3.15	1.98

Entropy fraction. Among Off and the three values of ρ, $\rho = 0.7$ is the best on seven of the eight columns. The gain is largest on the I2I benchmark, where an over-confident student under-edits, with the score rising from 2.04 to 2.31. The raw CFG target is higher only on Aesthetic, by 0.02. Softening further hurts. At $\rho = 0.9$, generation quality drops sharply, with GenEval at 0.176 and a negative ImageReward. This suggests that too much softening weakens the guided target.

Fixed temperature. Entropy-matched guidance solves for a temperature at each position and step. A simpler alternative divides the guided logits by one fixed temperature, $\nu = 1.4$ or $\nu = 2.0$.

Both settings score below Off on every benchmark we evaluate, and the drop grows with ν. At $\nu = 1.4$, GenEval falls from 0.798 to 0.475. At $\nu = 2.0$, GenEval falls to 0.015, DPG-Bench to 26.2 and ImageReward to -2.11. On the I2I benchmark, the two settings score 2.03 and 1.98, while entropy-matched guidance scores 2.31. A fixed temperature raises the entropy above the target late in decoding (Figure 10), which likely weakens the guided target. The gain of entropy-matched guidance thus comes from matching the entropy at each position, not from softening the target alone.