CONFLUX: SELF-REFINING UNIFIED DIFFUSION LLM WITH CROSS-PARADIGM ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose Conflux, a unified discrete diffusion large language model (DLLM) for visual understanding and generation tasks. Compared with prior unified diffusion models, Conflux introduces two major innovations. First, we propose a unified cross-paradigm on-policy distillation (OPD) framework that successfully transfers the visual understanding ability of autoregressive VLMs and visual generation ability of continuous flow models, to a unified DLLM. Second, Conflux incorporates a novel progressive self-refining diffusion process that gradually corrupts each token across multiple steps during training, as opposed to adopting the conventional masked diffusion formulation that directly corrupts a clean image token to an uninformative mask token. At inference, this design allows Conflux to make intermediate commit to a token, enabling progressive self-refinement for better visual quality. Through extensive experiments, we show that Conflux achieves state-of-the-art performance among unified DLLMs on diverse tasks including visual question-answering, multi-modal reasoning, object grounding, text-to-image generation, and image editing.

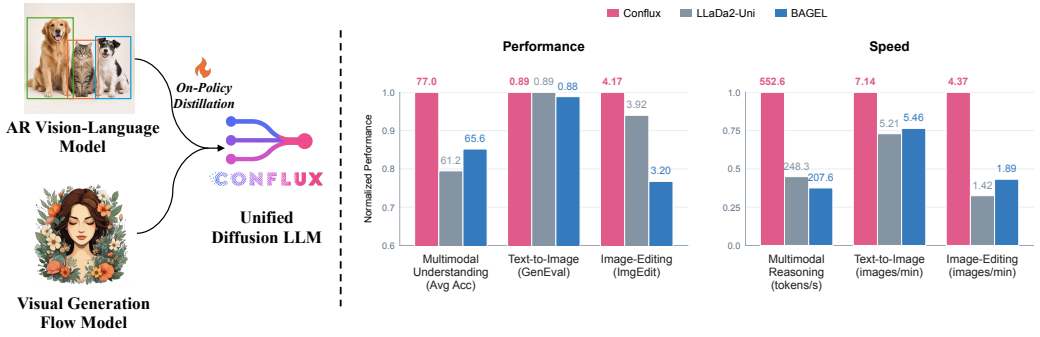


Figure 1: (Left) Conflux introduces a new paradigm for building unified Diffusion LLMs through on-policy distillation from cross-paradigm teachers: VLMs for visual understanding and continuous diffusion models for visual generation. (Right) Conflux achieves highly competitive visual understanding and generation performance. Visual understanding performance is averaged across tasks in Table 1. Text-to-image and image editing reports GenEval and ImgEdit Score. Multimodal reasoning is measured on MathVista. Axes are normalized by the best model's performance on each task.

1 INTRODUCTION

Unified Multimodal Large Language Models (MLLMs) (OpenAI, 2024; Gemini Team et al., 2024; Chen et al., 2025a;c) have achieved strong performance on diverse multimodal understanding and generation tasks. Compared with specialist models, they learn to perform diverse tasks via a unified sequence modeling interface, allowing synergetic learning that benefits each individual tasks (Deng et al., 2025). They generally fall into two categories. The first design represents text and image as discrete tokens and employs a unified autoregressive (AR) modeling paradigm (Lu et al., 2022a; Chen et al., 2025c; Wang et al., 2024b). The second design combines autoregressive modeling for

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.

text generation and continuous diffusion for image generation (AR+Diff) (Deng et al., 2025; Pan et al., 2025; Mo et al., 2025; NVIDIA, 2026). Several recent works such as LaViDa-O (Li et al., 2025b) and MMaDA (Yang et al., 2025) explored a third alternative that builds a unified discrete diffusion language model (unified DLLM) with masked diffusion paradigm (MDM). Unlike AR models or continuous flow models, MDMs randomly mask discrete tokens in a sequence and learn to predict the masked tokens during training. At inference, MDMs initialize a fully masked sequence and gradually unmask tokens across multiple diffusion steps. Compared with AR models, MDMs offer faster inference speed because they can generate multiple tokens in parallel (Sahoo et al., 2024; Shi et al., 2024a). Compared with AR+Diff setup, which needs to carefully balance conflicting AR and Diffusion objectives (Deng et al., 2025), unified DLLMs offer a single objective for both text and image, resulting in simplicity and, more importantly, scalability.

Despite these advantages, a substantial performance gap remains between existing unified DLLMs and domain-specialized VLMs and continuous flow models (Yang et al., 2025; Li et al., 2025b). In this work, we seek to narrow these gaps by introducing Conflux, a unified DLLM featuring a new training paradigm and new architecture.

First, although on-policy distillation (OPD) has proven highly effective in improving the visual understanding capabilities of VLMs (Liu et al., 2026b) and the visual generation capabilities of continuous flow models (Li et al., 2026a), adapting it to unified MDMs is challenging because of a lack of strong-performing MDM teachers. Distilling from strong AR VLMs and continuous flow models is not feasible because they follow fundamentally different paradigms. To overcome this challenge, we propose **cross-paradigm on-policy distillation (OPD)**, which enables Conflux to learn from teachers with different generation paradigms, including an AR VLM teacher for visual understanding and a continuous flow teacher for visual generation.

Second, for visual generation tasks specifically, MDMs lack the self-refinement capabilities of continuous flow models. At inference, each token jumps directly from a masked state to a clean state and stays unchanged afterwards. Since MDMs sample tokens independently from per-token logit prediction, all tokens sampled at the same diffusion step are assumed to be conditionally independent. This assumption rarely holds in practice since tokens from the same image are highly correlated. To address this limitation and enable iterative refinement, we introduce a **self-refining masked diffusion process** that incorporates new intermediate states between the clean state and mask state. By breaking down the one-step jump to a multi-step transition, Conflux enables fine-grained inter-token interaction and progressive self-correction, resulting in better visual quality.

To evaluate the effectiveness of Conflux, we conduct extensive evaluations on a diverse range of multimodal understanding and generation tasks, including visual question answering, object grounding, multimodal reasoning, text-to-image generation, and image editing. Experiments show that Conflux achieves state-of-the-art performance among unified DLLMs and greatly reduces the gap to specialized AR VLMs and continuous flow models.

The main contributions of this work are summarized as follows: 1) We develop a cross-paradigm OPD framework that improves both visual understanding and generation by distilling knowledge from non-DLLM teachers. 2) We introduce a self-refining masked diffusion process that improves visual generation quality over the canonical masked diffusion process. 3) We present Conflux, a state-of-the-art unified DLLM capable of both understanding and generating visual content. We acknowledge that while extensive experiments show the effectiveness of our proposed methods, the model can be further improved via better data mixture and architecture optimization.

2 Preliminaries and Related Works

2.1 Diffusion Language Models

Discrete diffusion language models (Lou et al., 2023; Sahoo et al., 2024; Nie et al., 2025) generate multiple text tokens in parallel, making them more efficient than AR models. However, its bidirectional attention introduces overhead compared to causal attention in AR models, particular for long sequences. Several works (Arriola et al., 2025; Cheng et al., 2026) address this limitation through block-wise semi-auto-regressive generation. Notably, SBD (Gat et al., 2025; Zhu et al., 2026) incorporates variable-sized blocks during training, enabling flexible inference.

Formally, given a block of clean text tokens B^0, the forward diffusion process gradually mask it over the continuous time interval $t \sim [0, 1]$ until it reaches a fully masked block B^1 containing only the mask token `[m]`. Given a multi-block token sequence $X = [B_1^0, B_2^0, ...B_N^0]$, SBD models learn to predict the clean block B_i^0 from its partially masked version B_i^t conditioned on all previous clean block. The optimization objective is defined as follows:

$$\mathcal{L}_{\text{SBD}} = \mathbb{E}_{i\sim\{1,2,..N\},t\sim[0,1],X\sim\mathcal{D}}[-\frac{1}{t}\log p(B_i^0|B_{j<i}^0, B_i^t)], \tag{1}$$

where $\mathcal{D}$ is the data distribution. At inference, the blocks are sampled sequentially, but tokens in the same block are sampled in parallel across multiple diffusion steps. Conflux applies the SBD paradigm to unified multimodal modeling and treats each image as an integral block, whose size is resolution-dependent. Note that when block size reduces to 1, SBD models reduce to an AR model.

2.2 Masked Discrete Image Models

Masked image models generate images by initializing a fully-masked image and gradually unmasking image patches. Each patch is encoded to a discrete code through vector quantization (VQ) (Esser et al., 2021). MaskGiT (Chang et al., 2022) pioneered this paradigm. Messonic (Bai et al., 2024) scaled it to large-scale T2I. Recent works such as LaViDa-O and MMaDa (Li et al., 2025b; Yang et al., 2025) explored integrating masked image modeling to unified MLLMs. As discussed in Sec. 1, these models lack the self-refinement capabilities of continuous flow models. Some work mitigate this problem using a specialized unmasking schedule that prioritizes unmasking less-correlated tokens (Besnier et al., 2025; Li et al., 2025b). Others explored revising already unmasked (Wang et al., 2025a; Li et al., 2026b). However, these works are sensitive to heuristic-based confidence thresholds and still present a mismatch between training and inference setup.

2.3 On-Policy Distillation (OPD)

OPD (Agarwal et al., 2024; Gu et al., 2024) enables capability transfers from a teacher model to a student model by providing teacher supervision signals on student-generated sequences. M-OPD (Ma et al., 2026) extends OPD to a multi-teacher setting to consolidate the capabilities of domain-specific experts. Several works (Yuan et al., 2026; Hao et al., 2026) explored OPD for multimodal understanding tasks. Diffusion-OPD (Li et al., 2026a) further applies it to continuous flow models for visual generation tasks. However, OPD for DLLMs is currently underexplored. OPDLM (Su et al., 2026) is the most relevant work, but its scope is limited to same-model distillation that converts a pure AR model to a DLLM with weaker performance but faster inference. Unlike prior work, we propose cross-paradigm OPD to jointly distill knowledge from a strong AR VLM teacher and an image diffusion generation teacher into a unified DLLM.

Beyond the three areas above, Conflux is closely related to broader literature on unified MLLMs, and we provide more discussion in Appendix E.1.

3 Method

3.1 Model Architecture

Conflux adopts a two-tower Elastic Mixture-of-Transformer (Elastic MoT) design following Lavida-O (Li et al., 2025b), which consists of an understanding expert tower and generation expert tower. They interact with each other through a joint attention mechanism. The understanding tower is initialized with a diffusion language model NLD (Fu et al., 2026b), which uses the block-causal SBD formulation described in Section 2.1. The generation tower is initialized from an identical copy. Following the common design of unified MLLMs (Deng et al., 2025; Li et al., 2025b), we encode images with both a semantic encoder and VQ encoder. Once images and texts are tokenized, the canonical SBD formulation can be directly applied. Figure 2 illustrates the overall framework.

3.2 Discrete Diffusion with Progressive Self-Refinement

The canonical mask discrete diffusion uses binary mask states, where a token directly jumps between the mask and clean states. Differently, our Conflux incorporates intermediate states using the

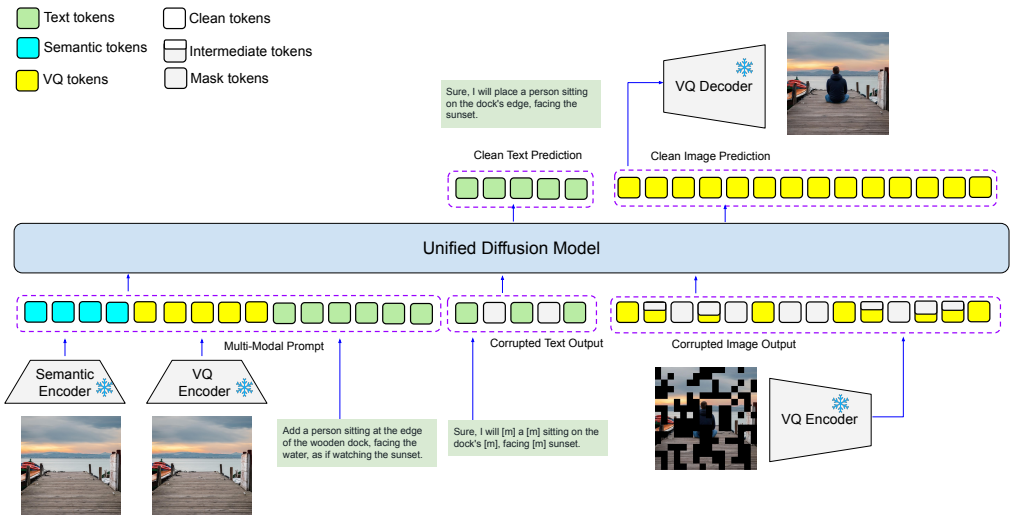


Figure 2: **Overall Framework of Conflux.** Conflux is a unified diffusion model whose input and output are both multimodal. Images in the input stream are represented by both continuous embeddings of a semantic encoder and discrete tokens of a VQ encoder. Images in the output stream are represented solely using VQ codes. Conflux is trained to predict the clean token distribution given partially masked inputs. For image generation, we incorporate an intermediate masking state.

concept of token age $a \in \mathcal{A} = \{0, 1, 2...K\}$. In particular, $a = 0$ implies a token is in the clean state, $a = K$ implies a token is in the fully-masked state. Crucially, a is a token-local discrete index and is different from the global continuous timestep $t \in [0, 1]$.

Formally, given a vocabulary $\mathcal{V}$, we adopt the state space $S = \mathcal{V} \times \mathcal{A}$ as opposed to $S = \mathcal{V} \cup \{[m]\}$ of standard MDMs. This is shown in Figure 3 (Left). We define the self-refining diffusion process as the interaction of a **discrete time per-token corruption process** and a **continuous time per-token jump schedule**. Specifically, each token in $(c, a) \in S$ can only transition to $(c', a + 1)$ in the forward process, with $(*, 0)$ being the initial states and $(*, K)$ being the terminal states. The transition probability is defined as $U_{k|k-1}(u|c)$, which defines the probability of a token $(c, k - 1)$ transits to (c, k). This generates a discrete time per-token corruption process where each token undergoes K jumps from initial clean state and terminal mask state. Given a token $X_i^0 = (C_i^0, A_i^0)$, the continuous time per-token jump schedule $b_i^t(a) = \mathbb{P}(A_i^t = a)$ determines when the jumps occur for each token in the interval $[0, 1]$. It satisfies $b_i^0(0) = 1$ and $b_i^1(K) = 1$ at the initial and terminal states.

Per token Corruption Process. The per-step transition rate defined above generates the marginal $U_{j|k}(u|c)$ with $0 \leq k \leq j \leq K$. It must satisfy $U_{K|k}(u|c) = Q(u)$, where $Q(u)$ is a stationary distribution over $\mathcal{V}$, for all $k < K$ because the terminal state is semantically fully masked and cannot correlate to the initial states. We implement this by randomly replacing a token c with a sampled token u in the VQ embedding space, where the jump distance depends on the token age. Concretely, we construct a balanced binary ball tree (Huang & Tung, 2023) which is commonly used in efficient nearest neighbor search. Each node in the ball tree represents a high-dimensional ball. in the embedding space, with parent nodes having a larger radius than child nodes. Let $N_d(c)$ be the set of all tokens that share a common ancestor to c at depth d, and D be the tree depth, it satisfies $N_1(c) = \mathcal{V}$ for the root node and $N_D(c) = \{c\}$ for the leaf node. An example of this structure is shown in Figure 3 (top-right), where we visualize a 4-depth tree.

We define $U_{k|k-1}(u|c)$ as the uniform distribution over $N_{D-k}(c)$, which gives the following forward and reverse marginal:

$$U_{j|k}(u|c) = \mathbf{1}\{u \in N_{D-j}(c)\}\frac{1}{|N_{D-j}(c)|}; \quad U_{k|j}(c|u, \hat{c}) = U_{k|0}(c|\hat{c}), \forall u \in N_{D-j}(\hat{c}) \tag{2}$$

where $\hat{c}$ is the initial token state at $a = 0$, and $0 \leq k < j \leq K$. LHS satisfies that the initial distribution is a one-hot distribution of the clean token and the terminal distribution is the uniform distribution over all masked states $(*, K)$. RHS can be used at inference by substituting the clean token $\hat{c}$ with a model prediction. We defer full derivation to Appendix A.2.1. Figure 3 (bottom-

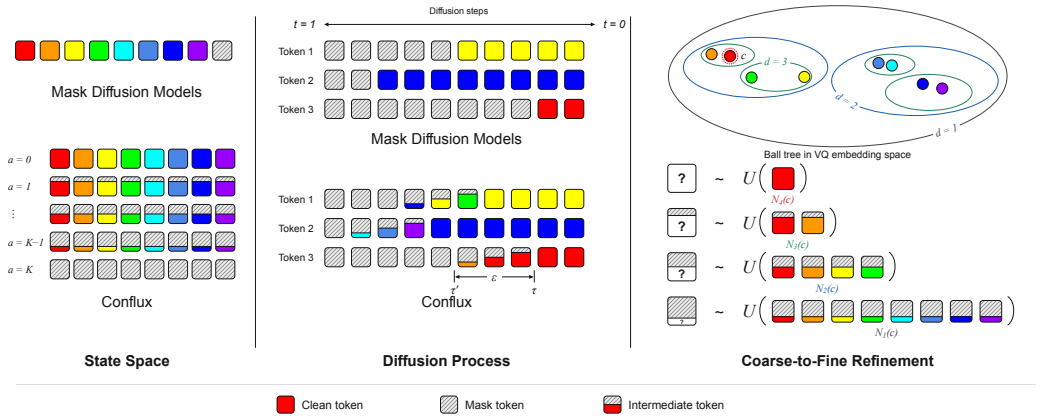


Figure 3: **Design of Self-Refinement Process.** (Left) We extend the state space of mask diffusion with token age a with $a = 0$ being the clean state and $a = K$ being the masked state. (Middle) Instead of one-step jump, each token is progressively masked through the interval (τ, τ') with length ϵ. Colors encode different tokens. (Right) We implement the progression from age $a = 0$ to $a = K$ through random jumps based on a ball-tree in the embedding space. Token distributions are diffused from a one-hot vector to balls with growing sizes. This example has tree depth $D = 4$.

right), illustrates the progression of marginal $U_{j|0}(\hat{c})$. It corrupts the token $\hat{c}$ to progressively random tokens in progressively larger balls in the embedding space, until it reaches the terminal state. We adopt the ball-tree mainly for efficiency. The full pairwise distance matrix would require 32GB VRAM since Conflux has 131k distinct VQ codes. By contrast, the corresponding 18-depth binary tree can be represented via a single 1D array taking only 524 KB VRAM.

Per-Token Jump Schedule embeds the discrete jumps in the continuous interval $[0, 1]$. We adopted DNDM (Chen et al., 2024)'s setup which generates the same forward marginal as vanilla MDM by first sampling a per-token jump time $\tau_i \sim U([0, 1])$ for each token i. The forward token-age transition then becomes deterministic with each token jumps from the clean state to the mask state exactly at $t = \tau_i$. We extend this to support multiple intermediate states by sampling the interval $(\tau_i, \tau_i') \subset [0, 1]$ via $\tau_i \sim U([0, 1-\epsilon])$ and set $\tau_i' = \tau_i + \epsilon$, where ϵ is a hyperparameter controlling the duration of intermediate states. When $\epsilon \to 0$, the process reduces to DNDM. When $\epsilon = 1$, all tokens are progressively corrupted at the same rate. To minimize the complexity, we let all intermediate token ages $1, 2, ..K-1$ to progress uniformly in (τ, τ'), taking $\frac{\epsilon}{K}$ time each. Given fixed $\epsilon \in (0, 1)$, this establishes a bijection between every instance of per-token jump schedule to a one-step jump schedule in DNDM. This process becomes non-Markov as we cannot marginalize τ_i. At training and inference, we presample (τ_i, τ_i') for each token, then the changes in token age become deterministic. The forward and reverse marginal can be obtained by directly applying Equation 2 with change-of-variable. Particularly, at inference, token i stays masked in $[\tau_i', 1]$, is progressively refined in $[\tau_i, \tau_i']$ and stays fixed in $[0, \tau_i]$. We defer additional details and derivations to Appendix A.2.2.

The most relevant work to Conflux in this aspect is MDM-prime (Chao et al., 2025), which breaks tokens into subtokens to enable partial masking. However, its image experiments are limited to 32x32 low-res images. We find that it does not scale to our setting. By contrast, Conflux makes use of codebook geometry to enable partial masking and unmasking in a coarse-to-fine manner, without the need to learn explicit sub-token representations. We provide more discussion in Appendix C.1.1.

3.3 Distillation of Visual Understanding Ability

To distill knowledge from AR teachers, we exploit SBD's ability to operate with varying block sizes. As described in Section 2.1, SBD models the conditional distribution $p(B_i^0 \mid B_1^0, \ldots, B_{i-1}^0, B_i^t)$ for a sequence of blocks $X = [B_1^0, \ldots, B_N^0]$, predicting each clean block from its corrupted version and the preceding clean blocks. When block size is one, SBD reduces to a standard AR model (AR-view). Hence, we can directly supervise the student's AR view with an AR teacher on student's AR

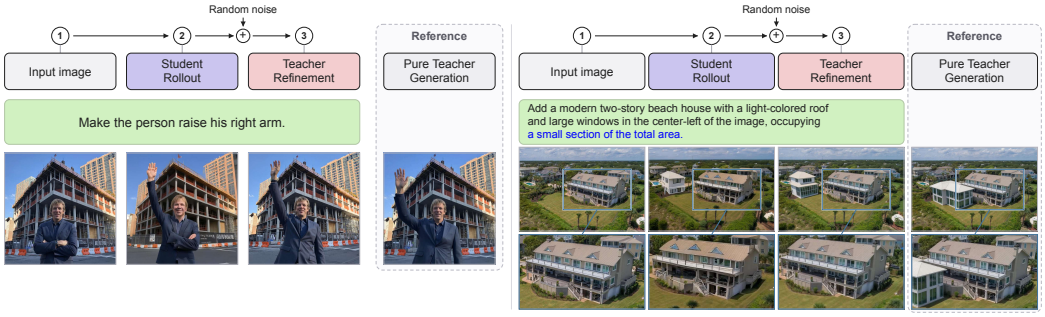


Figure 4: **On-Policy Distilling from Continuous Flow Models.** Given a student output, we generate a teacher-refined output by adding noise to the student sample and re-denoise it with the teacher model. For reference, we also show an independent teacher generation. We observe that the teacher's refinements are closely aligned to the student's output while having better visual fidelity and source image preservation.

rollouts. However, using AR OPD loss alone would lose the capability of generating multiple tokens without any supervision on the diffusion view.

A naive design is to directly copy the teacher score to the diffusion following OPD-LM (Su et al., 2026), however it creates ill-formed supervison signals due to context mismatch. As a concrete example, consider the student answer `"Jason visited San Francisco"` for a QA task. A teacher may want to respond `"Jason visited New York"` conditioned on the first two words and assigns high confidence to the token `"New"`. This supervision target is reasonable in AR setting. However, if we employ a block size of 2 and consider the partial state `"[m] Francisco"`, directly copying scores of AR teacher would guide the student generation towards generating `"New Francisco"`, which is ill-formed.

To address this challenge, we propose a transitive distillation objective that supervises the student's AR view with teacher scores and supervises the student's diffusion view with its own AR view. Specifically, we first generate a clean block B_i^0 and supervise it with AR OPD loss. To supervise the diffusion view, we apply the forward process and corrupt B_i^0 to a partially masked version B_i^t and minimize the KL divergence between the student's AR prediction and diffusion prediction. The final understanding loss can be defined as follows:

$$\mathcal{L}_{\text{OPD-und}} = \mathbb{D}_{\text{KL}}(p_{\text{s}}^{\text{AR}}(B_i^0|B_{j<i}^0)||p_{\text{t}}^{\text{AR}}(B_i^0|B_{j<i}^0)) + \mathbb{D}_{\text{KL}}(p_{\text{s}}^{\text{AR}}(B_i^0|B_{j<i}^0)||p_{\text{s}}^{\text{Diff}}(B_i^0|B_{j<i}^0, B_i^t)), \tag{3}$$

where p_s, p_t denote student and teacher distributions. The first term is the standard AR OPD loss, the second term is a self-consistency loss. We apply stop-grad for p_s^{AR} in consistency loss and only evaluate masked positions. Crucially, while the context is still different, the AR scores have good alignment because tokens in B_i^0 are sampled from p_s^{AR} with a low temperature.[1] We note that OPD-LM (Su et al., 2026) is applied to AR2Diff conversion where the student is initialized with the teacher weights. The teacher's gap to the student's AR-view is initially 0. Our setup distills an AR teacher that is larger and stronger to an already trained student diffusion model, which is more general but has a considerable teacher-student distribution gap at initialization, creating ill-formed signals when applying OPD naively. [2] We defer additional details to Appendix C.3.

3.4 Distillation of Visual Generation Ability

Distilling continuous flow models to DLLM is more challenging as flow models operate in continuous latent spaces, making exact KL-style distribution matching infeasible. However, the general notion of using teacher supervision on student-generated sequences is still feasible. In this work, we explore cross-paradigm OPD for image editing tasks with Qwen-Image (Wu et al., 2025a) teacher. Given a student output, we first re-encode it with the teacher's continuous VAE. We then create a

[1] We provide studies on sampling temperature in Appendix C.3

[2] OPD-LM also finds that for Qwen-8B, self-supervision outperforms Qwen-32B teacher in AR2Diff.

teacher supervision target by first adding noise to the student generation in teacher's latent space and re-denoise it with the teacher model, conditioned on the input image and instruction. We call this target teacher-refined student output. This pipeline is shown in Figure 4 (right). We observe that teacher-refined images are semantically and spatially aligned to student generation, but with reduced artifacts, higher fidelity, and better source-image preservation. By contrast, independent teacher generations exhibit a larger gap to student generation. This distinction makes the refined samples ideal candidates for training targets. We also found that teacher-refined outputs can be better than independent teacher generations where the teacher model makes mistakes while the student model does not. We defer more discussion and examples to Appendix A.4.

We adopt the canonical SFT loss on the teacher-refined samples. While this is not "on-policy" in the conventional sense, it constructs teacher-refined targets from the current student's own rollouts, adapting supervision to its evolving generation distribution. It closely connects to several concurrent OPD studies in AR language modeling (Jiang et al., 2026), which also explored using teacher models to correct student rollouts.

4 Experiment results

4.1 Main Results

Setup. We train Conflux in a three-stage manner. The first stage primarily includes text-image pairs for basic visual alignment and image generation capabilities. The second stage introduces additional data for instruction following, image editing, object grounding, and interleaved generation. The final stage is the unified OPD training stage, with Nemotron-3 Nano Omni as the teacher for multimodal reasoning tasks and Qwen-Image for image editing tasks. We also mix T2I SFT batches to preserve the text-to-image generation performance. Notably, we do not include an OPD teacher for T2I tasks because the base model's performance on GenEval and HPSv3 is stronger than teacher candidates like Qwen-Image and Flux. The training is conducted on 64 H100 GPUs over 26 days. In particular, the final OPD stage takes 5 days. We defer additional details to Appendix B.

Generic Understanding Tasks. We report the performance of image understanding tasks in Table 1. Conflux outperforms the previous state-of-the-art unified diffusion model and diffusion VLMs like LLaDa-V (You et al., 2025), NLD-VLM (Fu et al., 2026b) and LaViDa-O (Li et al., 2025b) by considerable margin on a wide range of benchmarks such as MMMU (Yue et al., 2024), ChartQA (Masry et al., 2022), InfoVQA (Mathew et al., 2022) and AI2D(Kembhavi et al., 2016). On object grounding benchmark RefCOCO (Kazemzadeh et al., 2014) and GUI grounding benchmark ScreenSpot-Pro (Li et al., 2025a), Conflux also demonstrates strong performance while most unified models are not equipped with this ability, unlocking new potential applications for diffusion language models. These gains are mostly attributed to scaling training data. Overall, Conflux reduces the gap between unified diffusion models and state-to-the-art AR models on visual understanding.

Multimodal Reasoning Tasks. We report the performance of multimodal reasoning tasks in Table 2 and compare Conflux with pre-OPD checkpoint (Conflux-Base). Our proposed cross-paradigm OPD framework improves performance on a wide range of tasks, with most notable improvements in We-Math (Qiao et al., 2025), Geometry3k (Lu et al., 2021), and MathVerse (Zhang et al., 2024). It also achieves state-of-the-art performance among diffusion language models. We additionally include a naive OPD baseline using OPD-LM (Su et al., 2026) style loss which directly copies teacher scores as the supervision target for diffusion generation, this worsens the performance of Conflux-Base because of the mismatch we identified in Section 3.3. These results highlight the effectiveness and necessity of the proposed cross-paradigm OPD framework.

Text-to-Image. We report results on GenEval (Ghosh et al., 2023), DPG (Hu et al., 2024), and HPS v3 (Ma et al., 2025) for a comprehensive evaluation. Conflux achieves strong results on these benchmarks and outperforms existing unified diffusion MLLMs such as MMaDA (Yang et al., 2025) and LaViDa-O(Li et al., 2025b). Most notably, on HPSv3 which measures visual quality, Conflux closes the gap between discrete diffusion models and continuous diffusion models and even surpasses strong baselines like Flux.1-dev (Black Forest Labs, 2024) and Qwen-Image (Wu et al., 2025a), highlighting the effectiveness of the self-refining process.

Table 1: **Quantitative results on image-level understanding tasks.** Conflux is evaluated using a confidence-based diffusion sampler following Wu et al. (2025b).

Model *Split*	MMMU val	RealWorldQA test	ChartQA test	DocVQA val	InfoVQA val	TextVQA val	AI2D test	RefCOCO val	Sci.QA test (image)	ScreenSpot-Pro test	OCRBenchV2 test (EN)	Avg
Autoregressive Models												
LLaVA-1.6-7B (Liu et al., 2024)	35.1	-	64.9	74.4	37.1	-	66.6	-	73.2	-	-	-
Qwen2.5-VL-7B (Bai et al., 2025b)	58.6	-	84.9	82.6	-	-	83.9	-	-	-	-	-
Qwen3-VL-8B (Bai et al., 2025a)	69.6	71.5	89.6	95.6	83.4	82.9	84.8	87.3	75.2	54.6	64.1	78.1
Cosmos3-Nano (NVIDIA, 2026)	60.9	72.2	85.5	94.5	81.5	83.4	85.0	79.2	93.0	56.1	65.9	77.9
BAGEL (Deng et al., 2025)	55.3	73.9	74.3	94.3	60.7	86.0	88.9	33.3	97.3	3.4	53.9	65.6
Discrete Diffusion Models												
Dimple (Yu et al., 2025)	45.2	-	63.4	-	-	-	74.4	-	77.1	-	-	-
LaViDa (Li et al., 2025c)	43.6	54.5	64.6	59.0	34.2	-	70.0	-	80.2	-	-	-
NLD-VLM (Fu et al., 2026b)	50.4	60.3	76.6	88.3	57.8	71.8	74.7	51.6	72.0	0.1	37.5	58.3
Fast-DVLM (Wu et al., 2026)	44.6	65.1	82.8	92.1	75.6	76.1	79.7	35.6	77.0	0.0	38.9	60.7
LLaDa-V (You et al., 2025)	48.6	63.2	78.3	83.9	66.3	-	77.8	-	0.0	-	-	-
MMaDa (Yang et al., 2025)	30.2	-	9.8	10.9	14.9	-	66.6	-	55.8	-	-	-
LaViDa-O (Li et al., 2025b)	45.1	-	80.0	73.7	44.6	-	76.7	91.2	84.6	0.0	-	-
LLaDA2-Uni (AI et al., 2026)	50.1	66.7	80.1	89.5	70.1	70.0	82.0	39.9	88.2	0.4	36.0	61.2
Conflux	58.4	68.2	89.1	93.9	80.8	77.7	84.2	94.1	92.3	46.7	61.3	77.0
Δ vs. prev. Diffusion SOTA	***+8.0***	***+1.5***	***+6.3***	***+1.8***	***+5.2***	***+1.6***	***+2.2***	***+2.9***	***+4.1***	***+46.3***	***+22.4***	***+15.8***

Table 2: **Quantitative results on multi-modal reasoning benchmarks.** *Denotes naively applying teacher score to student's diffusion view using OPD-LM's loss.

Model	MathVista	DynaMath	MathVision	MathVerse	Geometry3k	LogicVista	We-Math
Autoregressive Models							
Qwen2.5-VL-7B (Bai et al., 2025b)	68.2	55.7	22.4	39.0	37.1	42.4	33.3
Qwen3-VL-8B (Bai et al., 2025a)	77.2	62.2	37.9	67.7	57.5	50.1	69.0
BAGEL (Deng et al., 2025)	73.1	-	24.1	-	-	-	45.8
Discrete Diffusion Models							
Fast-dVLM (Wu et al., 2026)	57.3	51.3	20.2	33.7	27.0	37.1	41.0
NLD-VLM (Fu et al., 2026b)	56.0	48.2	21.5	28.5	29.3	38.3	50.1
LLaDA2-Uni (AI et al., 2026)	67.2	52.7	19.3	32.8	35.1	31.5	48.0
LaViDa-O (Li et al., 2025b)	54.0	36.1	13.8	27.0	23.5	34.2	34.5
Conflux-Base	71.5	68.3	35.8	35.5	54.4	52.4	59.8
+ Naive OPD*	47.7	46.7	17.9	33.4	23.0	38.7	42.0
+ Cross-Paradigm OPD (Conflux)	75.0	71.8	38.6	41.2	68.2	55.5	68.1
Δ *vs. Base*	*+3.5*	*+3.5*	*+2.8*	*+5.7*	*+13.8*	*+3.1*	*+8.3*

Table 3: **Text-to-Image Generation Performance on GenEval, DPG, and HPSv3.** † LLaDa-2-Uni incorporates a pretrained 6B continuous diffusion decoder. *Half of the understanding layers are not loaded in T2I tasks due to the Elastic-MoT design.

Model	Family	Params	GenEval							DPG	HPSv3
			Single Obj.↑	Two Obj.↑	Counting↑	Colors↑	Position↑	Color Attri.↑	Overall↑	Overall↑	Score↑
Gen. Only											
SD3-Medium (Esser et al., 2024)	Cont.	2B	0.99	0.94	0.72	0.89	0.33	0.60	0.74	84.1	-
FLUX.1-dev (Black Forest Labs, 2024)	Cont.	12B	0.98	0.81	0.74	0.79	0.22	0.45	0.66	83.8	10.88
Qwen-Image(Wu et al., 2025a)	Cont.	20B	0.99	0.92	0.89	0.88	0.76	0.77	0.87	88.3	10.91
Meissonic (Bai et al., 2024)	Disc.	1B	0.99	0.66	0.42	0.86	0.10	0.22	0.54	-	-
NLD-Image(Li et al., 2026b)	Disc.	8B	0.98	0.93	0.83	0.94	0.88	0.82	0.90	85.2	9.61
Unified MLLM											
LLaDa-2-Uni (AI et al., 2026)	Hybrid	16B+6B †	1.00	0.98	0.73	0.92	0.90	0.84	0.89	87.8	9.47
BAGEL (Deng et al., 2025)	Cont.	14B	0.98	0.95	0.84	0.95	0.78	0.77	0.88	-	-
Emu3 (Wang et al., 2024b)	Disc.	8B	-	-	-	-	-	-	0.66	80.6	-
Janus-Pro (Chen et al., 2025c)	Disc.	7B	0.99	0.89	0.59	0.90	0.79	0.66	0.80	84.1	8.81
MMaDA (Yang et al., 2025)	Disc.	8B	0.99	0.76	0.61	0.84	0.20	0.37	0.63	53.4	5.43
Show-o (Xie et al., 2024)	Disc.	1.3B	0.98	0.80	0.66	0.84	0.31	0.50	0.68	-	7.20
LaViDa-O (Li et al., 2025b)	Disc.	10B	0.99	0.85	0.71	0.86	0.65	0.58	0.77	81.8	8.81
Conflux	Discrete.	12B*	0.99	0.96	0.79	0.87	0.90	0.82	0.89	86.6	11.12

Image-Editing. We report ImgEdit benchmark (Ye et al., 2025) in Table 4. Conflux outperforms discrete diffusion models like LaViDa-O (Li et al., 2025b) and RL-post-trained LaViDa-R1(Li et al., 2026c). Compared with a boarder range of model families, Conflux only underperforms GPT-4o and the Qwen-Image teacher, greatly reducing the gap between discrete diffusion models and the state-of-the-art works. Compared with pre-OPD checkpoint Conflux-Base, our cross-paradigm OPD leads to substantial improvements (+0.36). We also evaluate an off-policy KD baseline which supervises the student with independent teacher-generated samples. Our OPD framework, which uses teacher-refined student samples, outperforms the offline setup by a considerable margin (+0.27). We provide more discussion in the appendix.

4.2 Ablation Studies

Self-Refining Diffusion Process. We compare with the uniform diffusion, masked diffusion, and masked diffusion with token editing (Li et al., 2026b) on GenEval, DPG, and HPSv3 in Figure

Table 4: **Per-Category and overall scores on ImgEdit benchmark.** *Off-policy indicates we use an independent teacher output as the target, as opposed to teacher-refined student output.

Model	Add	Adjust	Extract	Replace	Remove	Action	Background	Style	Hybrid	Overall
Other Methods										
Qwen-Image (Wu et al., 2025a)	4.38	4.16	3.43	4.66	4.14	4.69	4.38	4.81	3.82	4.27
GPT-4o (OpenAI, 2024)	4.61	4.33	2.90	4.35	3.66	4.89	4.57	4.93	3.96	4.20
LLaDa2-Uni (AI et al., 2026)	3.76	4.16	2.40	4.04	3.82	4.42	4.07	4.60	3.97	3.92
Qwen2.5VL+Flux (Wang et al., 2025b)	4.07	3.79	2.04	4.13	3.89	4.52	3.90	4.84	3.04	3.80
FluxKontext dev (Black Forest Labs et al., 2025)	3.76	3.45	2.15	3.98	2.94	4.26	3.78	4.38	2.96	3.52
OmniGen2 (Wu et al., 2025c)	3.57	3.06	1.77	3.74	3.20	4.68	3.57	4.81	2.52	3.44
BAGEL (Deng et al., 2025)	3.56	3.31	1.70	3.30	2.62	4.17	3.24	4.49	2.38	3.20
Step1X-Edit (Liu et al., 2025)	3.88	3.14	1.76	3.40	2.41	2.52	3.16	4.63	2.64	3.06
InstructAny2Pix(Li et al., 2023)	2.55	1.83	2.10	2.54	1.17	1.98	2.01	3.51	1.42	2.12
Instruct-Pix2Pix(Brooks et al., 2023)	2.45	1.83	1.44	2.01	1.50	1.46	1.44	3.55	1.20	1.88
Discrete Diffusion Models										
LaViDa-O (Li et al., 2025b)	4.04	3.62	2.01	4.39	3.98	3.54	4.06	4.82	2.94	3.71
LaViDa-R1(Li et al., 2026c)	4.25	3.90	2.32	4.52	4.06	4.18	3.86	4.87	3.10	3.90
Conflux-Base	4.11	4.02	2.26	4.13	3.83	4.19	4.06	4.49	3.23	3.81
+ Off-Policy KD*	4.13	4.15	2.45	4.51	3.97	4.21	4.21	4.53	2.97	3.90
+ Cross-Paradigm OPD (Conflux)	4.26	4.33	3.59	4.63	4.20	4.06	4.31	4.64	3.48	4.17

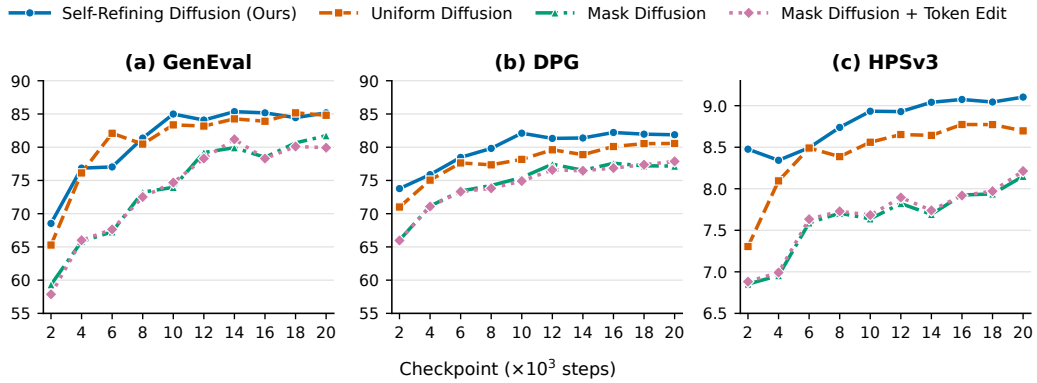


Figure 5: **Ablation Studies on the Self-Refining Diffusion Process.** We report performance on GenEval (a) DPG (b) and HPSv3 (c) benchmarks at different training steps using different paradigm.

5. We use 64 diffusion steps for all evaluations. We show the performance of models are trained with each method throughout a 20k-step run. On average, the proposed self-refining diffusion process performs the best. Conflux outperforms masked diffusion overall. Compared with uniform diffusion,Conflux is comparable on GenEval, which checks high-level semantics using an object detector. On more fine-grained benchmarks like DPG and human-preference-aligned benchmarks like HPSv3, Conflux consistently outperforms the uniform baseline. Among all benchmarks, the improvements on HPSv3 is the most significant, suggesting that the improvements of self-refinement is best for improving visual fidelity.

Cross-Paradigm On-policy Distillation. In Table 2 and 4, we compare our proposed OPD framework with a naive OPD baseline on multimodal reasoning and an off-policy KD baseline on image editing tasks. We provide more experiment studies, such as the design of each loss term, the relation between AR performance and diffusion performance, and the effect of noise ratio in teacher refinement step to Appendix C.

5 Conclusion

Diffusion language models have attracted considerable attention in recent years for their greater efficiency compared with autoregressive models and their unified interface for multimodal data. However, their performance in practice still lags behind that of state-of-the-art models, limiting their adoption. In this paper, we propose two methods for developing more capable DLLMs: cross-paradigm on-policy distillation, which transfers knowledge from diverse teacher models, and a self-refining diffusion process, which improves visual generation quality. Extensive experiments demonstrate the effectiveness of both approaches. We hope Conflux will offer valuable insights for future research on DLLMs and unified multimodal systems.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic datasets, clean and reformat datasets, implement methods, support qualitative and thematic data analysis, and process results. We did not use generative AI tools to develop theoretical models or conceptual frameworks, design or provide feedback on research methodology or experiments, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in writing proofs, propose or refine hypotheses, or assist with translation. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

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. *arXiv preprint arXiv:2503.09573*, 2025. URL https://arxiv.org/abs/2503.09573.

Jinbin Bai, Tian Ye, Wei Chow, Enxin Song, Xiangtai Li, Zhen Dong, Lei Zhu, and Shuicheng Yan. Meissonic: Revitalizing masked generative transformers for efficient high-resolution text-to-image synthesis. *arXiv preprint arXiv:2410.08261*, 2024. URL https://arxiv.org/abs/2410.08261.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*, 2025b. URL https://arxiv.org/abs/2502.13923.

Victor Besnier, Mickael Chen, David Hurych, Eduardo Valle, and Matthieu Cord. Halton scheduler for masked generative image transformer. *arXiv preprint arXiv:2503.17076*, 2025. URL https://arxiv.org/abs/2503.17076.

Black Forest Labs. FLUX, 2024. URL https://github.com/black-forest-labs/flux.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. FLUX.1 Kontext: Flow Matching for In-Context Image Generation and Editing in Latent Space, 2025. URL https://arxiv.org/abs/2506.15742.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.

Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. COYO-700M: Image-Text Pair Dataset, 2022. URL https://github.com/kakaobrain/coyo-dataset.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11315–11325, 2022.

Chen-Hao Lance Chao, Wei-Fang Sun, Hanwen Liang, Chun-Yi Lee, and Rahul Krishnan. Beyond masked and unmasked: Discrete diffusion models via partial masking. *Advances in Neural Information Processing Systems*, 38:97458–97500, 2025.

Guanzhou Chen, Erfei Cui, Changyao Tian, Danni Yang, Ganlin Yang, Yu Qiao, Hongsheng Li, Gen Luo, and Hongjie Zhang. Scaleedit-12m: Scaling open-source image editing data generation via multi-agent framework. *arXiv preprint arXiv:2603.20644*, 2026.

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025a. URL https://arxiv.org/abs/2505.09568.

Junying Chen, Zhenyang Cai, Pengcheng Chen, Shunian Chen, Ke Ji, Xidong Wang, Yunjin Yang, and Benyou Wang. ShareGPT-4o-Image: Aligning Multimodal Models with GPT-4o-Level Image Generation. *arXiv preprint arXiv:2506.18095*, 2025b. URL https://arxiv.org/abs/2506.18095.

Lin Chen and Long Xing. Open-LLaVA-NeXT: An open-source implementation of LLaVA-NeXT series for facilitating the large multi-modal model community., 2024. URL https://github.com/xiaoachen98/Open-LLaVA-NeXT.

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. URL https://arxiv.org/abs/2501.17811.

Zixiang Chen, Huizhuo Yuan, Yongqian Li, Yiwen Kou, Junkai Zhang, and Quanquan Gu. Fast sampling via discrete non-markov diffusion models with predetermined transition time. *Advances in Neural Information Processing Systems*, 37:106870–106905, 2024.

Shuang Cheng, Yihan Bian, Dawei Liu, Yuhua Jiang, Yihao Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, et al. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 22058–22075, 2026.

Yufeng Cui, Honghao Chen, Haoge Deng, Xu Huang, Xinghang Li, Jirong Liu, Yang Liu, Zhuoyan Luo, Jinsheng Wang, Wenxuan Wang, et al. Emu3. 5: Native multimodal models are world learners. *arXiv preprint arXiv:2510.26583*, 2025.

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. URL https://arxiv.org/abs/2505.14683.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Ling Fu, Zhebin Kuang, Jiajun Song, Mingxin Huang, Biao Yang, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, et al. Ocrbench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning. *Advances in Neural Information Processing Systems*, 38, 2026a.

Yonggan Fu, Lexington Whalen, Abhinav Garg, Chengyue Wu, Maksim Khadkevich, Nicolai Oswald, Enze Xie, Daniel Egert, Sharath Turuvekere Sreenivas, Shizhe Diao, et al. Nemotron-Labs-Diffusion: A Tri-Mode Language Model Unifying Autoregressive, Diffusion, and Self-Speculation Decoding. *arXiv preprint arXiv:2607.05722*, 2026b. URL https://arxiv.org/abs/2607.05722.

Itai Gat, Heli Ben-Hamu, Marton Havasi, Daniel Haziza, Jeremy Reizenstein, Gabriel Synnaeve, David Lopez-Paz, Brian Karrer, and Yaron Lipman. Set block decoding is a language model inference accelerator. *arXiv preprint arXiv:2509.04185*, 2025. URL https://arxiv.org/abs/2509.04185.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. URL https://arxiv.org/abs/2403.05530.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge Distillation of Large Language Models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Jarvis Guo, Tuney Zheng, Yuelin Bai, Bo Li, Yubo Wang, King Zhu, Yizhi Li, Graham Neubig, Wenhu Chen, and Xiang Yue. Mammoth-vl: Eliciting multimodal reasoning with instruction tuning at scale. *arXiv preprint arXiv:2412.05237*, 2024. URL https://arxiv.org/abs/2412.05237.

Dongze Hao, Zhiwei Jin, Chen Chen, and Haonan Lu. Stabilizing On-Policy Distillation for MLLM Reasoning with Global Normalization. *arXiv preprint arXiv:2606.09091*, 2026. URL https://arxiv.org/abs/2606.09091.

Greg Heinrich, Mike Ranzinger, Hongxu Yin, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. RADIOv2.5: Improved Baselines for Agglomerative Vision Foundation Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22487–22497, 2025.

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.

Qiang Huang and Anthony KH Tung. Lightweight-yet-efficient: Revitalizing ball-tree for point-to-hyperplane nearest neighbor search. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 436–449. IEEE, 2023.

Yiming Jia, Jiachen Li, Xiang Yue, Bo Li, Ping Nie, Kai Zou, and Wenhu Chen. VisualWebInstruct: Scaling up Multimodal Instruction Data through Web Search. *arXiv preprint arXiv:2503.10582*, 2025. URL https://arxiv.org/abs/2503.10582.

Li Jiang, Haoran Xu, Yichuan Ding, and Amy Zhang. Trajectory-refined distillation. *arXiv preprint arXiv:2606.08432*, 2026.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 787–798, 2014.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pp. 235–251. Springer, 2016.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario, Canada, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. URL https://arxiv.org/abs/2408.03326.

Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. Screenspot-pro: Gui grounding for professional high-resolution computer use. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 8778–8786, 2025a.

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, 2026a. URL https://arxiv.org/abs/2605.15055.

Shufan Li, Harkanwar Singh, and Aditya Grover. Instructany2pix: Flexible visual editing via multimodal instruction following. *arXiv preprint arXiv:2312.06738*, 2023. URL https://arxiv.org/abs/2312.06738.

Shufan Li, Jiuxiang Gu, Kangning Liu, Zhe Lin, Zijun Wei, Aditya Grover, and Jason Kuen. Lavida-O: Elastic Masked Diffusion Models for Unified Multimodal Understanding and Generation. *arXiv preprint arXiv:2509.19244v1*, 2025b. URL https://arxiv.org/abs/2509.19244v1. Version 1.

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. *arXiv preprint arXiv:2505.16839*, 2025c. URL https://arxiv.org/abs/2505.16839.

Shufan Li, Greg Heinrich, Hanrong Ye, Yonggan Fu, Aditya Grover, Jan Kautz, and Pavlo Molchanov. Nemotron-Labs-Diffusion-Image: Advancing Masked Discrete Diffusion for High-Resolution Image Synthesis. *arXiv preprint arXiv:2606.29814*, 2026b. URL https://arxiv.org/abs/2606.29814.

Shufan Li, Yuchen Zhu, Jiuxiang Gu, Kangning Liu, Zhe Lin, Yongxin Chen, Molei Tao, Aditya Grover, and Jason Kuen. LaViDa-R1: Advancing reasoning for unified multimodal diffusion language models. *arXiv preprint arXiv:2602.14147*, 2026c. URL https://arxiv.org/abs/2602.14147.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024. URL https://arxiv.org/abs/2411.04996.

Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, et al. UniWorld: High-Resolution Semantic Encoders for Unified Visual Understanding and Generation. *arXiv preprint arXiv:2506.03147v1*, 2025. URL https://arxiv.org/abs/2506.03147v1. Version 1.

Alexander H Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, et al. Ministral 3. *arXiv preprint arXiv:2601.08584*, 2026a.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge, January 2024. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Ruiqi Liu, Xiaolei Lv, Gengsheng Li, Ximo Zhu, Zhiheng Wang, Zhengbo Zhang, Junkai Chen, Zhiheng Li, Bo Li, Jun Gao, et al. Visual-advantage on-policy distillation for vision-language models. *arXiv preprint arXiv:2605.21924*, 2026b. URL https://arxiv.org/abs/2605.21924.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025. URL https://arxiv.org/abs/2504.17761.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023. URL https://arxiv.org/abs/2310.16834.

Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. *arXiv preprint arXiv:2206.08916*, 2022a.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6774–6786, 2021.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022b.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023. URL https://arxiv.org/abs/2310.02255.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, Jinhao Dong, Zhifang Sui, and Fuli Luo. MOPD: Multi-Teacher On-Policy Distillation for Capability Integration in LLM Post-Training, 2026. URL https://arxiv.org/abs/2606.30406.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15086–15095. IEEE, 2025.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177. URL https://aclanthology.org/2022.findings-acl.177.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2200–2209, 2021.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1697–1706, 2022.

Sicheng Mo, Thao Nguyen, Xun Huang, Siddharth Srinivasan Iyer, Yijun Li, Yuchen Liu, Abhishek Tandon, Eli Shechtman, Krishna Kumar Singh, Yong Jae Lee, et al. X-fusion: Introducing new modality to frozen large language models. *arXiv preprint arXiv:2504.20996*, 2025. URL https://arxiv.org/abs/2504.20996.

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. *arXiv preprint arXiv:2502.09992*, 2025. URL https://arxiv.org/abs/2502.09992.

Yifan Niu, Han Xiao, Dongyi Liu, Zelong Wang, Dihong Gong, Yasheng Wang, and Jia Li. Breaking the Tokenizer Barrier: On-Policy Distillation across Model Families, 2026. URL https://arxiv.org/abs/2606.09456.

NVIDIA. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026. URL https://research.nvidia.com/labs/cosmos-lab/cosmos3/technical-report.pdf.

NVIDIA, Amala Sanjay Deshmukh, Kateryna Chumachenko, Tuomas Rintamaki, Matthieu Le, Tyler Poon, Danial Mohseni Taheri, Ilia Karmanov, Guilin Liu, Jarno Seppanen, Arushi Goel, et al. Nemotron 3 nano omni: efficient and open multimodal intelligence. *arXiv preprint arXiv:2604.24954*, 2026. URL https://arxiv.org/abs/2604.24954.

OpenAI. GPT-4o System Card. *arXiv preprint arXiv:2410.21276*, 2024. URL https://arxiv.org/abs/2410.21276.

Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, et al. Transfer between modalities with metaqueries. *arXiv preprint arXiv:2504.06256*, 2025. URL https://arxiv.org/abs/2504.06256.

Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma Gongque, Shanglin Lei, Yifan Zhang, et al. We-math: Does your large multimodal model achieve human-like mathematical reasoning? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20023–20070, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M. Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S. Khan. GLaMM: Pixel Grounding Large Multimodal Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13009–13018, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Rasheed_GLaMM_Pixel_Grounding_Large_Multimodal_Model_CVPR_2024_paper.html.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Christoph Schuhmann. LAION-AESTHETICS, 2022. URL https://laion.ai/blog/laion-aesthetics/. Accessed: 2024 - 03 - 06.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

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*, 2024a.

Weijia Shi, Xiaochuang Han, Chunting Zhou, Weixin Liang, Xi Victoria Lin, Luke Zettlemoyer, and Lili Yu. LMFusion: Adapting Pretrained Language Models for Multimodal Generation. *arXiv preprint arXiv:2412.15188*, 2024b. URL https://arxiv.org/abs/2412.15188.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA Models That Can Read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8317–8326, 2019.

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, 2026. URL https://arxiv.org/abs/2606.06712.

Keqiang Sun, Junting Pan, Yuying Ge, Hao Li, Haodong Duan, Xiaoshi Wu, Renrui Zhang, Aojun Zhou, Zipeng Qin, Yi Wang, et al. Journeydb: A benchmark for generative image understanding. *Advances in neural information processing systems*, 36:49659–49678, 2023.

Guanghan Wang, Yair Schiff, Subham Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. *Advances in Neural Information Processing Systems*, 38:147282–147339, 2025a.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring Multimodal Mathematical Reasoning with MATH-Vision Dataset. In *The Thirty-eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024a. URL https://openreview.net/forum?id=QWTCcxMpPA.

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*, 2024b. URL https://arxiv.org/abs/2409.18869.

Yuhan Wang, Siwei Yang, Bingchen Zhao, Letian Zhang, Qing Liu, Yuyin Zhou, and Cihang Xie. GPT-IMAGE-EDIT-1.5M: A Million-Scale, GPT-Generated Image Dataset. *arXiv preprint arXiv:2507.21033*, 2025b. URL https://arxiv.org/abs/2507.21033.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025a. URL https://arxiv.org/abs/2508.02324.

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*, 2025b. URL https://arxiv.org/abs/2505.22618.

Chengyue Wu, Shiyi Lan, Yonggan Fu, Sensen Gao, Jin Wang, Jincheng Yu, Jose M Alvarez, Pavlo Molchanov, Ping Luo, Song Han, et al. Fast-dvlm: Efficient block-diffusion vlm via direct conversion from autoregressive vlm. *arXiv preprint arXiv:2604.06832*, 2026.

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. OmniGen2: Exploration to Advanced Multimodal Generation. *arXiv preprint arXiv:2506.18871v1*, 2025c. URL https://arxiv.org/abs/2506.18871v1. Version 1.

xAI. RealWorldQA Dataset, 2024. URL https://huggingface.co/datasets/xai-org/RealworldQA.

Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. Logicvista: Multimodal llm logical reasoning benchmark in visual contexts. *arXiv preprint arXiv:2407.04973*, 2024.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024. URL https://arxiv.org/abs/2408.12528.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. MMaDA: Multimodal Large Diffusion Language Models. *arXiv preprint arXiv:2505.15809*, 2025. URL https://arxiv.org/abs/2505.15809.

Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. *arXiv preprint arXiv:2505.20275*, 2025. URL https://arxiv.org/abs/2505.20275.

Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. Llada-v: Large language diffusion models with visual instruction tuning. *arXiv preprint arXiv:2505.16933*, 2025. URL https://arxiv.org/abs/2505.16933.

Runpeng Yu, Xinyin Ma, and Xinchao Wang. Dimple: Discrete diffusion multimodal large language model with parallel decoding. *arXiv preprint arXiv:2505.16990*, 2025. URL https://arxiv.org/abs/2505.16990.

Qianhao Yuan, Jie Lou, Xing Yu, Hongyu Lin, Le Sun, Xianpei Han, and Yaojie Lu. Vision-opd: Learning to see fine details for multimodal llms via on-policy self-distillation. *arXiv preprint arXiv:2605.18740*, 2026. URL https://arxiv.org/abs/2605.18740.

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 *Proceedings of CVPR*, 2024.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, et al. MathVerse: Does Your Multi-modal LLM Truly See the Diagrams in Visual Math Problems? *arXiv preprint arXiv:2403.14624*, 2024. URL https://arxiv.org/abs/2403.14624.

Yuchen Zhu, Jing Shi, Chongjian Ge, Hao Tan, Yiran Xu, Wanrong Zhu, Jason Kuen, Koustava Goswami, Rajiv Jain, Yongxin Chen, et al. FLARE: Diffusion for Hybrid Language Model. *arXiv preprint arXiv:2606.01774*, 2026. URL https://arxiv.org/abs/2606.01774.

Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models. In *International Conference on Learning Representations*, volume 2025, pp. 48337–48383, 2025.

A Additional Technical Details

A.1 Formulations of SBD Models

This section briefly describes the formulation of SBD models. Conflux largely adopts its canonical form.

Given a sequence of discrete tokens $X^0 = [X_1^0, X_2^0, ..X_L^0]$, AR models generate tokens by repetitively sampling from the next-token prediction $p(X_i^0|X_{j<i}^0)$. By contrast, masked discrete diffusion models (MDMs) initialize a fully masked sequence $X^1 = [[m], [m], ...]$ and gradually unmask tokens during the continuous interval $[0, 1]$ through the reverse process $p(X^s|X^t)$, where $0 \leq s \leq t \leq 1$, until the sequence reaches a clean state X^0 that consists of only non-mask tokens. At each step, tokens are sampled independently by assuming the factorization $p(X^s|X^t) = \prod_i p(X_i^s|X^t)$. Since $p(X_i^s|X^t)$ depends on the entire sequence X^t for every token position i, MDMs employ bidirectional attention over the full sequence as opposed to causal attention in AR models. Consequently, MDMs suffer from efficiency bottlenecks despite being capable of generating multiple tokens in parallel.

To address this limitation, models like Block Diffusion (Arriola et al., 2025), SDAR (Cheng et al., 2026) and SBD (Gat et al., 2025) adopt a semi-autoregressive generation paradigm.

Specifically, the full sequence $X^0 = [X_1^0, X_2^0, ..X_L^0]$ is broken down into multiple consecutive multi-token blocks $X^0 = [B_1^0, ...B_K^0]$. The tokens within each block are generated with the standard MDM sampling process, but the blocks are sampled sequentially from left to right. This design only requires within-block bidirectional attention, greatly reducing the compute cost of fully bidirectional MDMs.

Formally, let $X_{b,i}^t$ denote the i-th token in the b-th block. The forward diffusion process is defined as

$$q(X_{b,i}^t|X_{b,i}^s) = \begin{cases} \mathbf{M}, & \text{if } X_{b,i}^s = [M] \\ \frac{1-t}{1-s}\mathbf{X_{b,i}^s} + \frac{t-s}{1-s}\mathbf{M}, & \text{if } X_{b,i}^s \neq [M], \end{cases} \tag{4}$$

where $0 \leq s \leq t \leq 1$, and $\mathbf{M}, \mathbf{X_{b,i}^s}$ are one-hot probability vectors corresponding to the mask token `[m]` and the clean token $X_{b,i}^s$.

When $s = 0$, we obtain the following marginal:

$$q(X_{b,i}^t|X_{b,i}^0) = (1-t)\mathbf{X_{b,i}^0} + t\mathbf{M}. \tag{5}$$

The posterior of the reverse process $p(X^s|X^t, X^0)$ has the following form (Sahoo et al., 2024):

$$p(X_{b,i}^s|X_{b,i}^0, X_{b,i}^t) = \begin{cases} \mathbf{X_{b,i}^t} & \text{if } X_{b,i}^t \neq [M] \\ \frac{t-s}{t}\mathbf{X_{b,i}^0} + \frac{s}{t}\mathbf{M}, & \text{if } X_{b,i}^t = [M]. \end{cases} \tag{6}$$

.

Practically, a neural network θ is used to model the clean token prediction $p_\theta(X_{b,i}^0|B_{j<b}^0, B_b^t)$ conditioned on the partially masked current block B_b^t and all previous clean blocks $B_{j<b}^0$, which gives the following transition:

$$p_\theta(X_{b,i}^s|B_{j<b}^0, B_b^t) = \begin{cases} \mathbf{X_{b,i}^t} & \text{if } X_{b,i}^t \neq [M] \\ \frac{t-s}{t}p_\theta(X_{b,i}^0|B_{j<b}^0, B_b^t) + \frac{s}{t}\mathbf{M}, & \text{if } X_{b,i}^t = [M]. \end{cases} \tag{7}$$

.

Training Process. SBD employs a specialized attention mask to allow parallel training over multiple blocks. Given a sequence $X^0 = [X_1^0, X_2^0, ..X_N^0]$ sampled from some data distribution $\mathcal{D}$ and a randomly sampled timestep $t \sim U([0, 1])$, we first divide X^0 into consecutive blocks $B_1^0, ...B_K^0$ and construct corresponding partially masked blocks $B_1^t, ...B_K^t$ using the forward marginal in Equation 5. The input to the model is constructed by concatenating the clean sequence and partially masked sequence, obtaining $[B_1^0, ...B_K^0, B_1^t, ...B_K^t]$. A specialized attention mask is used to ensure the correct semantics and allow for parallel multi-block training. In particular, tokens in the clean block B_i^0 can attend to preceding clean tokens in B_i^0 and all tokens in preceding clean blocks $B_{j<i}^0$. Tokens in

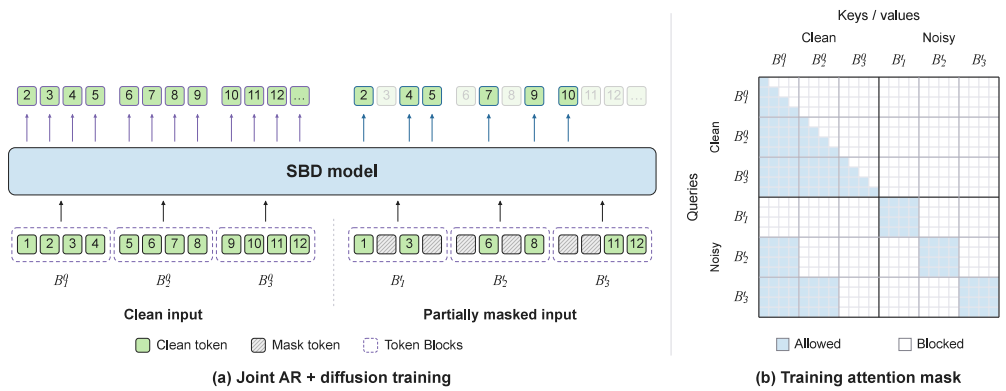


Figure 6: **The design of SBD formulation.** (Left) SBD adopts a two-stream training process that concatenates clean blocks and noisy blocks in the sequence dimension. It applies AR CE loss on the clean blocks, and diffusion CE loss on the noisy blocks. (Right) SBD models use a specialized attention mask with a causal mask in the clean blocks and block-bidirectional mask in the noisy blocks. Each noisy block can attend to all other tokens in itself and all tokens in preceding clean blocks.

the partially masked block B_i^t can attend to all tokens in the same partially masked block B_i^t and all tokens in preceding clean blocks $B_{j<i}^0$. The loss target is simply the concatenated clean sequence $Y = X^0|X^0$ shifted by 1 position to the left. This design is illustrated in Figure 6.

At each training step, the block size is randomized. The loss is also applied on the causal clean token portion of the input, which corresponds to a block size of one and is effectively an AR view. We note that Conflux adopts a position-shifted variant of SBD (Zhu et al., 2026) to better align the AR-view and diffusion view with a block size larger than 1, as opposed to the vanilla version Gat et al. (2025) which does not shift tokens on the partially masked portion of the input and has a mismatch between the AR view and the block-size-one diffusion view.

In pure language modeling, a sequence is divided into equally sized blocks. Conflux further allows varying-sized blocks in the same sequence because it ensures tokens from the same image belong to the same block and can attend to each other.

Sampling Process. At inference, we sample blocks $B_1, B_2..$ sequentially until some block contains a termination token. Each block is initialized with a fully masked block B_i^1 and progressively unmasked via the parameterized reverse process in Equation 7. In particular, we discretize the time interval [0,1] into discrete steps $0 = t_0 < t_1 < t_2..t_T = 1$, and recursively query Equation 7 starting from B_i^1 until we reach the clean state B_i^0.

A.2 Formulations of Self-Refining Diffusion Process

For text blocks, we employ the standard MDM formulation described in Section A.1 to minimize the gap with the base LLM NLD used to build Conflux. For image blocks, we employ our proposed self-refining diffusion process with intermediate mask states. For reduced complexity and sampling efficiency, we adopt the non-Markov formulation (DNDM) (Chen et al., 2024). For simplicity of notation, this discussion focuses on the setup with a single image sequence $X^0 = [X_1^0, ...X_L^0]$ whose tokens belong to one block. However, it can be easily extended to multi-block settings with interleaved multi-modal sequences.

A.2.1 Ball-Tree-Based Per-Token Corruption Process

Recall from Section 3.2 that we adopt the state space $\mathcal{V} \times \mathcal{A}$ where $\mathcal{V}$ is the vocabulary of a tokenizer and $\mathcal{A} = \{0, 1, ...K\}$ represents the set of corruption levels that connect the fully masked state and the clean state. Each token X_i^t in this state space is denoted as $X_i^t = (C_i^t, A_i^t)$ where $C_i^t \in \mathcal{V}$, $A_i^t \in \mathcal{A}$. In particular, $A_i = 0$ represents a clean state and $A_i = K$ represents a mask state. We

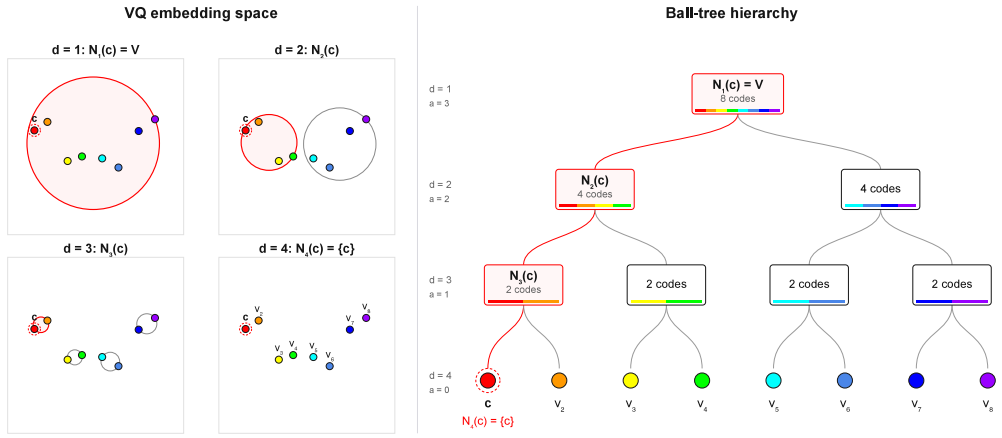


Figure 7: **Visualization of Ball-Tree-Based Corruption Process.** (Right) Ball-Tree is a balanced binary tree whose leaf nodes correspond to unique tokens in the vocabulary. The forward corruption process $U_{k|k-1}$ increases the token age a and decreases the node depth d. (Left) Each node has a geometric representation which is a high-dimensional ball in the VQ embedding space. Decreasing depth d corresponds to increasing ball radius. In this visualization, the depth $D = 4$ and maximum token age $K = 3$.

define the age-dependent corruption process such that A_i monotonically increases at state transitions in the forward process.

Concretely, each token that starts at state (c, a) for some $c \in \mathcal{V}, a \in \mathcal{A}$ can only transition to $(c', a+1)$, with $(*, 0)$ being the set of initial states and $(*, K)$ being the set of absorbing mask states. The corruption process can be expressed by the transition probability $U_{j|k}(u|c)$ for $0 \leq k < j \leq K$. It must satisfy the constraint:

$$U_{K|k}(u|c) = Q(u), \tag{8}$$

which implies that the absorbing state Q is an uninformative stationary distribution over possible mask states that is not dependent on the initial clean token c. Ideally we want the transition to be progressive so that the state becomes gradually less informative about the clean token c when we progress from $a = 0$ to $a = K$.

We implement this as a ball tree-based corruption process. A ball tree is a binary tree structure commonly used for fast nearest-neighbor search. The leaf nodes represent individual VQ codes in the continuous embedding space of the tokenizer; each non-leaf node represents a high-dimensional ball. The root node represents a ball enclosing all VQ codes. Child nodes represent a smaller ball than the parent node.

Concretely, given a D-depth ball tree with 2^{D-1} leaf nodes corresponding to tokens in $\mathcal{V}$, let $N_d(c)$ be the set of all tokens that share a common ancestor with c at depth d where $1 \leq d \leq D$. Notably, $N_1(c) = \mathcal{V}$ is the whole vocabulary and $N_D(c) = \{c\}$ contains exactly one code.

We set $K = D - 1$ as the total per-token corruption steps and associate them with $d = D - a$, with $a = 0$ corresponding to $d = D$ and $a = K$ corresponding to $d = 1$. Concretely, we generate $U_{j|k}$ with:

$$U_{k|k-1}(u|c) = \mathbf{1}\{u \in N_{D-k}(c)\} \frac{1}{|N_{D-k}(c)|}, \tag{9}$$

which is a uniform probability vector over the set $N_{D-k}(c)$, i.e., we jump from code c to a random neighboring token in the embedding space indicated by the ball tree, with the jump range dependent on the target token age. This process has the following marginal:

$$U_{j|k}(u|c) = \mathbf{1}\{u \in N_{D-j}(c)\} \frac{1}{|N_{D-j}(c)|}, \tag{10}$$

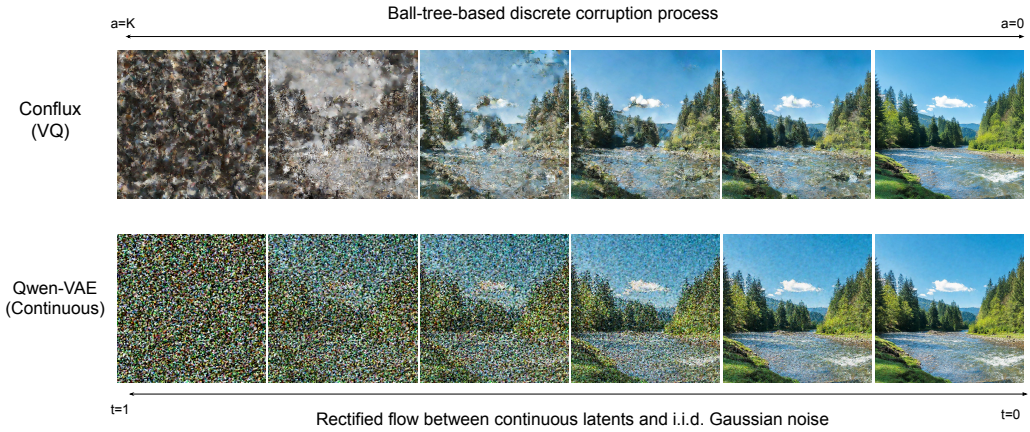


Figure 8: **Visual Reconstruction Across Different Token Ages.** To investigate whether the proposed age-based corruption process is semantically meaningful for visual generation tasks, we show reconstructed images at different token ages and compare them with the rectified flow process in the continuous latent space of Qwen-VAE. Results indicate that changes in the token age represent a smooth corruption process in the pixel space. In this visualization, all tokens are corrupted at the same rate, which corresponds to $\epsilon = 1$.

for $j > k$ because the intermediate states are always leaf nodes of $N_{D-j}(c)$ by tree construction. An important observation is that for a fixed u, $U_{j|k}(u|c)$ is either 0 or $\frac{1}{2^j}$ for all possible $c \in \mathcal{V}$.

Let $\hat{c}$ be the initial token state at $a = 0$. The conditional reverse process has the following form according to Bayes' rule:

$$U_{k|j}(c|u, \hat{c}) = \frac{U_{k|0}(c|\hat{c})U_{j|k}(u|c)}{U_{j|0}(u|\hat{c})}. \tag{11}$$

Note that for this value to be properly defined, we must have $U_{j|0}(u|\hat{c}) \neq 0$, implying $u \in N_{D-j}(\hat{c})$. When $U_{j|k}(u|c) \neq 0$, by the previously discussed observation the two terms cancel and the expression simplifies to $U_{k|0}(c|\hat{c})$. When $U_{j|k}(u|c) = 0$, it can be implied that $u \notin N_{D-j}(c)$. Hence $N_{D-j}(c) \neq N_{D-j}(\hat{c})$, and consequently $N_{D-k}(c) \neq N_{D-k}(\hat{c})$ because k, which satisfies $k < j$, corresponds to more refined tree nodes with greater depth. This implies that $U_{k|0}(c|\hat{c}) = 0$, so everywhere we have the reverse process satisfying:

$$U_{k|j}(c|u, \hat{c}) = U_{k|0}(c|\hat{c}) = \mathbf{1}\{c \in N_{D-k}(\hat{c})\}\frac{1}{|N_{D-k}(\hat{c})|}. \tag{12}$$

An implementation detail worth mentioning is that a model p_θ may still predict $\hat{c}$ such that $u \notin N_{D-j}(\hat{c})$ as its output space is the entire vocabulary $\mathcal{V}$. In those cases, we still apply the RHS by allowing the model to correct its previous predictions. In practice, we find that Conflux rarely predicts a token that violates this setting, and the constrained setup shows no meaningful difference from the unconstrained setup.

Figure 7 illustrates an example of this process with $D = 4$. On the left side, we visualize the set of leaf nodes $N_d(c)$ for $d = 1, 2, 3, 4$ corresponding to the token c (encircled with a dashed line) in the embedding space, which are neighboring tokens to c at different radii. On the right side, we show a binary tree representation of the same structure. The forward process starts with token age $a = 0$ and ends with age $a = K$, which corresponds to traversing to parent nodes in the binary tree from the leaf node at $d = 4$ to the root node at $d = 1$. In the embedding space, it interpolates the one-hot distribution of a single embedding c to progressively larger high-dimensional balls, until it reaches the full space.

Empirical Implementation. The tokenizer of Conflux contains 131k discrete codes, which corresponds to a ball tree with $D = 18$. Hence, the vanilla setup is $K = 17$. To enable flexible inference,

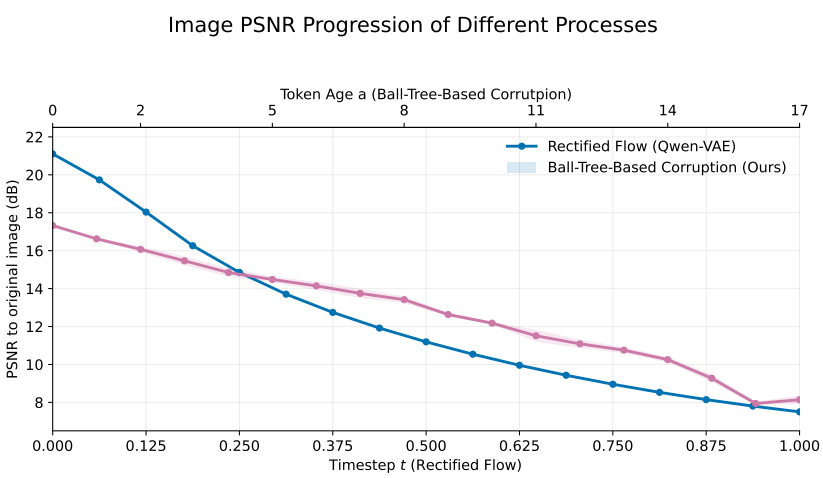


Figure 9: **Image PSNR Across Different Token Ages.** To investigate whether the proposed age-based corruption process is semantically meaningful for visual generation tasks, we plot the PSNR of reconstructed images at different token ages and compare them with the rectified flow process in the continuous latent space of Qwen-VAE. Results indicate that changes in the token age represent a smooth corruption process in the pixel space measured by PSNR. In the clean state, Qwen-VAE has better PSNR than Conflux, suggesting its continuous VAE has better reconstruction capability.

we in practice randomize K for each training step among the candidates $\{1, 2, ..17\}$, where for $K < 17$, we skip intermediate ages uniformly. For example, $K = 4$ corresponds to age progression $0, 4, 8, 12, 17$ in the original $K = 17$ notation.

To verify that the ball-tree-based corruption process is semantically meaningful, we visualize the progression of corrupt tokens in Figure 8 and compare it with the continuous rectified-flow setup. Specifically, we encode the images with Conflux's tokenizer and visualize the codes sampled from the defined corruption process $U_{k|k-1}(u|c)$. We also encode the images with the continuous VAE of Qwen-Image Wu et al. (2025a) and interpolate their latents with i.i.d. Gaussian noise following the rectified flow formulation. Visual results show that both processes gradually corrupt the semantics of the original image in a smooth manner.

We further confirm this observation quantitatively by measuring the PSNR against the ground truth image. The results are shown in Figure 9. The plot demonstrates that both processes offer a smooth transition from a clean image to random noise measured by PSNR. We observe that rectified flow has higher PSNR at zero corruption; this is attributed to the better reconstruction capability of Qwen-VAE as opposed to the corruption process.

As a final remark, we clarify that $U_{j|k}$ is a discrete-time per-token corruption process defined over discrete token ages $0, 1...K$, it should not be confused with the continuous-time diffusion process $q(X^t|X^s)$ with continuous timesteps $t, s \in [0, 1]$. The two processes are bridged by the token age field, which determines when discrete jumps $U_{k-1|k}$ occur over the continuous time interval $[0, 1]$ for each token.

A.2.2 Non-Markov Per-Token Jump Schedule

An important observation in the formulation described in Section A.1 is that during training, only the marginal:

$$q(X_i^t|X_i^0) = (1-t)\mathbf{X_i^0} + t\mathbf{M}, \tag{13}$$

is used to construct the partially masked sequence X^t and we never explicitly invoke $q(X_i^t|X_i^s)$ from Equation 4 for $s \neq 0$. Discrete non-Markov diffusion models (DNMD) (Chen et al., 2024) constructs a process that gives the same forward marginal with deterministic token transition time.

Specifically, the forward process can be defined as:

$$q(X_i^t|X_i^s,\tau_i) = \mathbf{1}\{t \le \tau_i\}\mathbf{X_i^s} + \mathbf{1}\{t > \tau_i\}\,\mathbf{M}, \tag{14}$$

where $\tau_i \sim U([0,1])$ is the clean-to-mask transition time that is pre-sampled once for each token and maintained throughout the trajectory. This design relaxes the Markov constraint of the process.[3] However, it is compatible with standard MDM training because it does not change the forward marginal or the clean token target distribution $p(X_i^0|X^t)$.

We make an additional remark that DNDM's setup in fact better aligns with the empirical sampling algorithms Chang et al. (2022); Besnier et al. (2025); Li et al. (2025b) which often predefine an unmasking schedule, whereas the exact Markov reverse process would unmask a stochastic number of tokens, or even zero tokens, at each step. Most recent samplers for masked image models like the Halton sampler (Besnier et al., 2025), and the stratified sampler (Li et al., 2025b) actually precompute the exact timestep to unmask each token, making DNDM a more accurate formulation in the image generation domain.

To extend this setup to allow intermediate transition states between the clean state and the mask state, we augment the generating process by first sampling a predefined transition period $[\tau_i, \tau_i']$ as opposed to a fixed jump time τ_i. In particular, we sample

$$\tau_i \sim U([0, 1-\epsilon]), \tau_i' = \tau_i + \epsilon, \tag{15}$$

where $\epsilon \in (0,1]$ is the hyper-parameter controlling the length of the transition process from clean state with $a=0$ to the mask state with $a=K$. In particular, when $\epsilon \to 0$, all jumps happen in an infinitesimal time interval and the overall process reduces to the canonical DNDM.

With a pre-sampled interval, the token age A_i can be expressed as a deterministic step function

$$a_i(t,\tau_i,\epsilon) = \begin{cases} 0 & \text{if} \quad 0 \le t \le \tau_i \\ \text{Floor}(K\frac{t-\tau_i}{\epsilon}) & \text{if} \quad \tau_i < t < \tau_i' \\ K & \text{if} \quad \tau_i' \le t \le 1 \end{cases}. \tag{16}$$

This induces a one-hot categorical distribution over the token age A_i^t:

$$b_i^t(a) = \mathbb{P}(A_i^t = a|\tau_i,\epsilon) = \mathbf{1}\{a = a_i(t,\tau_i,\epsilon)\}. \tag{17}$$

Hence, the forward process conditioned on a pre-sampled interval is:

$$q\big(X_i^t = (u,a) \mid X_i^s = (c,a'),\tau_i\big) = b_i^t(a)\begin{cases} U_{a|a'}(u \mid c), & a > a', \\ \mathbf{1}\{u=c\}, & a = a', \\ 0, & a < a'. \end{cases}. \tag{18}$$

The posterior reverse process is simply

$$p\big(X_i^s = (c,a') \mid X_i^t = (u,a), X_i^0 = (\hat{c},0),\tau_i\big) = b_i^s(a')\begin{cases} U_{a'|a}(c \mid u,\hat{c}), & a' < a, \\ \mathbf{1}\{c=u\}, & a' = a, \\ 0, & a' > a. \end{cases}, \tag{19}$$

where $U_{a|a'}(u|c)$ and $U_{a'|a}(c|u,\hat{c})$ are the forward and reverse marginals of the discrete-time per-token corruption process. At inference, X_i^0 is sampled from the model's clean token prediction.

A.3 On-policy Distillation from AR VLMs

The standard OPD framework is defined over AR language models sharing the same vocabulary. Specifically, given a teacher model and student model, the OPD loss (Gu et al., 2024) is defined as:

$$\mathcal{L}_{\text{AR-OPD}} = \mathbb{E}_{X^0 \sim p_{\text{student}}}[\frac{1}{L}\sum_{i=1}^{L}\mathbb{D}_{\text{KL}}(p_{\text{student}}(X_i^0|X_{j<i}^0)||p_{\text{teacher}}(X_i^0|X_{j<i}^0))], \tag{20}$$

[3] For the single-jump construction with a distinct absorbing mask token, the process remains Markov after marginalizing τ because the current state reveals whether the jump has occurred. More general single-jump DNDM can be non-Markov. For example, suppose the initial token is replaced by an independent uniform draw from $\mathcal{V}$ at a presampled time τ, and then held fixed.

where $X^0 = [X_1^0 ... X_L^0]$ is a sequence sampled from the student model. Recall from Section A.1 that the SBD model divides X^0 into consecutive blocks $X^0 = [B_1^0, ..B_K^0]$ and implements

$$p_\theta(B_{b,i}^0 | B_{j<b}^0, B_b^t) \tag{21}$$

where B_b^t is a partially masked version of the clean block B_b^0 and $B_{b,i}^0$ is the i-th token in the b-th block.

When the block size is 1, each block only has two states: a clean token and a mask token, so it is generated in one step by evaluating

$$p_\theta(B_{b,1}^0 | B_{j<b}^0, B_b^1) = p_\theta(B_{b,1}^0 | B_{j<b}^0) = p_\theta(X_b^0 | X_{j<b}^0) \tag{22}$$

where B_b^1 can be marginalized since there is only one possible state with a probability of 1. The final expression exactly implements an AR model, and we refer to this setup as the AR view of the SBD model. Practically, this distribution can be evaluated for all tokens in a sampled sequence in a single pass because of the shift-by-one logit implementation illustrated in Figure 6. Given this AR generation mode and AR view, we can directly apply the OPD loss used in AR models from Equation 20 to the SBD model with no change.

However, if we only sample from and supervise the model in its AR view, the SBD model would lose its support for parallel generation with block sizes larger than 1. To provide generic supervision for Equation 21 with a block size larger than 1, we define the AR-Diffusion Consistency loss as:

$$\mathcal{L}_{\text{Cons}} = \mathbb{E}_{X^0 \sim p_{\text{student}}, t \sim U([0,1])} [\frac{1}{KNt} \sum_{b=1}^{K} \sum_{i=1}^{N} \mathbb{D}_{\text{KL}}(p_{\text{student}}(B_{b,i}^0 | B_{j<b}^0, B_{b,r<i}^0) || p_{\text{student}}(B_{b,i}^0 | B_{j<b}^0, B_b^t))], \tag{23}$$

where $N > 1$ is the block size and B_b^t is a partially masked block sampled from the forward marginal $q(B_b^t | B_b^0)$. X^0 is sampled with AR mode with block size 1. The final loss for visual understanding and reasoning tasks is:

$$\mathcal{L}_{\text{OPD-UND}} = \mathcal{L}_{\text{AR-OPD}} + \mathcal{L}_{\text{Cons}}. \tag{24}$$

A.3.1 Connection with OPD-LM

In our notation convention, OPD-LM (Su et al., 2026)'s objective can be represented as a single term:

$$\mathcal{L}_{\text{OPD-LM}} = \mathbb{E}_{X^0, X^t \sim p_{\text{student}}} [\frac{1}{KN} \sum_{b=1}^{K} \sum_{i=1}^{N} \mathbb{D}_{\text{KL}}(p_{\text{teacher}}(B_{b,i}^0 | B_{j<b}^0, B_{b,r<i}^0) || p_{\text{student}}(B_{b,i}^0 | B_{j<b}^0, B_b^t))]. \tag{25}$$

Compared with our consistency loss, the main differences are

- OPD-LM samples rollouts from the diffusion view while Conflux samples from the AR view
- OPD-LM directly uses teacher scores to supervise the student's diffusion view, while Conflux uses the student's own AR score to supervise the diffusion view
- OPD-LM initializes the student with the teacher's weights, as an untrained diffusion model. In cross-paradigm OPD of Conflux, the teacher and the student are two distinct models with different sizes and architectures. The student model already has strong diffusion generation capabilities before OPD.

At the system level, the broader difference is that Conflux uses OPD as a component in a multi-objective unified training run combining visual understanding, text-to-image, and image editing objectives, while OPD-LM focuses solely on single-objective AR2Diff post-training in the language domain.

One of the problems with OPD-LM's setup is that it creates effectively ill-formed supervision signals due to potentially conflicting contexts. Specifically, high-probability tokens in

$p_{\text{teacher}}(B^0_{b,i}|B^0_{j<b}, B^0_{b,r<i})$ may not be good candidates semantically for $p_{\text{student}}(B^0_{b,i}|B^0_{j<b}, B^t_b)$ with a different context.

As a concrete example, consider the following student generation "`The number` i `is called the multiplicative identity.`". Given the prefix "`The number` i `is called the`", the teacher may want to generate the term for "`imaginary unit`", and assign high scores to the token "`imaginary`", but it would not be a good candidate if the masked context is "`The number` i `is called the [m] identity`" because the term "`imaginary identity`" is ill-formed.

The consistency loss of Conflux mitigates this issue by using the student's own AR predictions as supervision targets, avoiding the additional teacher–student distribution discrepancy while retaining the difference between AR and diffusion contexts. At zero sampling temperature, each token in a greedy rollout is an argmax of the student's AR conditional distribution, although this does not guarantee compatibility of the full supervision distribution with the observed future context. Empirically, we find that sampling AR rollouts at a small nonzero temperature yields better diffusion performance, suggesting that increased trajectory diversity can be beneficial despite relaxing the argmax property.

In Table 2 of the main paper, we show that directly applying OPD-LM's loss with heterogeneous teacher and student models performs poorly because of the mismatch identified above. We provide more analysis of each individual component in Section C.3.

A.4 On-policy Distillation from continuous flow models

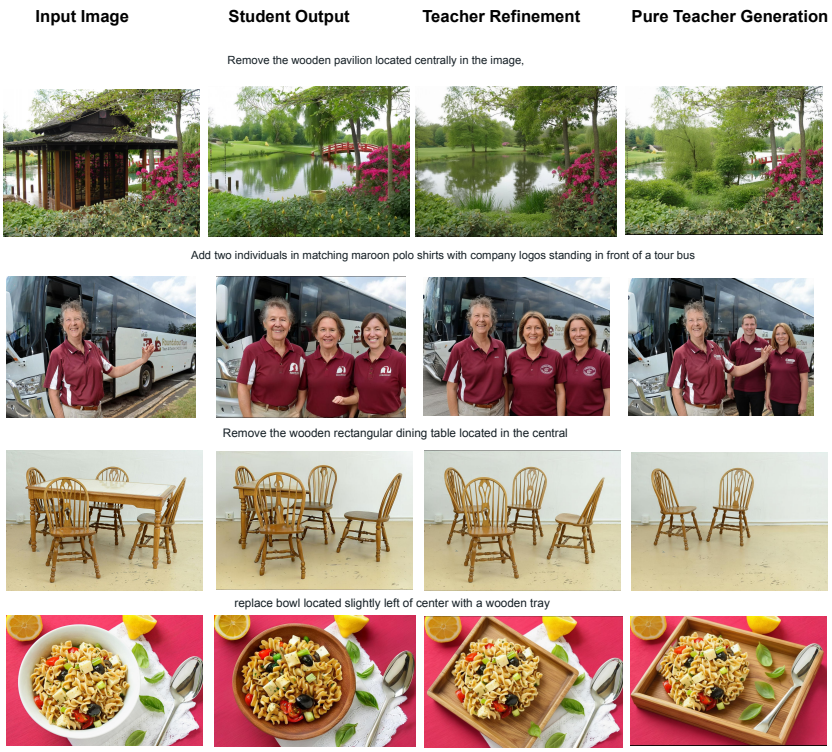


Figure 10: **Additional Qualitative Illustrations of the Teacher Refinement Process.** We show additional qualitative samples of teacher-refined student outputs. They generally align with the student roll-outs but have better visual fidelity and source image preservation. In some examples where the teacher over-edits, the refined student outputs are preferable to pure teacher generation.

In distilling from Qwen-Image on image editing tasks, we start with an input image V_{in} and text instructions p. We then generate discrete codes X^0_{out} by sampling from the self-refining diffusion

process $p_\theta(X^0_{\text{out}}|V_{\text{in}}, p)$. We decode them with Conflux's VQ decoder to obtain the image in pixel space $V_{out} = \text{Decode}_{\text{VQ}}(X_{\text{out}})$. We then re-encode V_{out} with the teacher's continuous VAE $x^0 = \text{Encode}_{\text{VAE}}(V_{\text{out}})$, and then create a partially noised version of the latent $x^r = (1-r)x^0 + r\delta$ where $\delta \sim N(0, I)$ is i.i.d. Gaussian noise and r is a hyperparameter controlling refinement strength. The teacher model $p_{\text{teacher}}(\hat{x}^0|x^r, p, V_{\text{in}})$ is employed to sample the clean prediction $\hat{x}^0$. Crucially, this refinement process involves multiple ODE steps as opposed to a single-step teacher forward pass.

The predicted latent $\hat{x}^0$ is decoded into pixel space and re-encoded into discrete VQ codes to obtain $\hat{X}^0 = \text{Encode}_{\text{VQ}}(\text{Decode}_{\text{VAE}}(\hat{x}^0))$. We then sample the partially corrupted sequence $\hat{X}^t$ using Equation 18. The final OPD objective is defined by:

$$\mathcal{L}_{\text{OPD-Gen}}(\hat{X}^0, \hat{X}^t, V_{\text{in}}, p, t) = -\frac{1}{t} \log p_\theta(\hat{X}^0|\hat{X}^t, V_{\text{in}}, p), \tag{26}$$

which is effectively the SFT loss on the refined target. The full process is described in Algorithm 1.

Algorithm 1 Teacher-Refined Distillation from On-Policy Student Rollouts

Require: Student p_θ; frozen teacher T; training data $\mathcal{D}$; refinement strength $r \in (0, 1)$; ODE steps $M > 1$; corruption-time distribution ρ on $(0, 1]$

while not converged **do**
  Sample a minibatch $\mathcal{B}$ from $\mathcal{D}$
  $\mathcal{L} \leftarrow 0$
  for $(V_{\text{in}}, p) \in \mathcal{B}$ **do**
    $X^0_{\text{out}} \sim p_\theta(\cdot \mid V_{\text{in}}, p)$ ▷ Current student rollout
    $V_{\text{out}} \leftarrow \text{Decode}_{\text{VQ}}(X^0_{\text{out}})$
    $x^0 \leftarrow \text{Encode}_{\text{VAE}}(V_{\text{out}})$
    $\delta \sim \mathcal{N}(0, I)$
    $x^r \leftarrow (1-r)x^0 + r\delta$
    $\hat{x}^0 \leftarrow \text{ODESolve}_T(x^r;\, r \to 0,\, M,\, V_{\text{in}}, p)$ ▷ Multi-step teacher refinement
    $\hat{X}^0 \leftarrow \text{stopgrad}\big(\text{Encode}_{\text{VQ}}\big(\text{Decode}_{\text{VAE}}(\hat{x}^0)\big)\big)$
    Sample $t \sim \rho$
    Sample $\hat{X}^t$ by corrupting $\hat{X}^0$ using Eq. 18
    $\mathcal{L} \leftarrow \mathcal{L} - \frac{1}{t} \log p_\theta(\hat{X}^0 \mid \hat{X}^t, V_{\text{in}}, p)$
  end for
  Update θ
end while

Connection to Trajectory-refined Distillation in Language Modeling. We note that while our objective is an SFT-style loss, the learning process is on-policy because the supervision target is derived from the student output. This paradigm closely connects with several concurrent works in the OPD literature in the language domain that explored student-teacher interleaved generation. Notably, Trajectory-refined Distillation (Jiang et al., 2026) proposes to use a teacher model to correct student trajectories in language tasks. Conflux expands this concept across model families and to the visual generation domain to address a lack of strong MDM teachers in visual generation tasks.

Qualitative Assessment. In addition to Figure 4, we provide additional samples in Figure 10 to illustrate the effectiveness of our proposed approach. We make two observations. First, compared with pure teacher generation, the teacher-refined student outputs are visually aligned to the student output but with more refined details and better source-image preservation, making them a more ideal training target. Second, we find that the teacher-refined student output can actually be better than the student model's outputs when the pure teacher generation makes a mistake. In the example "`Remove the wooden rectangular dining table`", teacher-refined student generation achieves the best results and outperforms both student and teacher generation. This occurs because when the teacher makes a mistake, the pure teacher generation tends to drift from the student example considerably (in this case, over-editing and removing two extra chairs in addition to the table). The refinement process, which constrains the teacher output to a neighborhood of the student roll-out, serves as useful regularization in these cases.

Table 5: **Training configurations across three stages.** We use letters A-C to represent different dataset.

	Stage 1	**Stage 2**	**Stage 3**
Learning Rate	1×10^{-5}	1×10^{-5}	2×10^{-6}
Steps	400k	100k $\times$ 2	10k
β_1	0.99	0.99	0.99
β_2	0.999	0.999	0.999
optimizer	AdamW	AdamW	AdamW
Dataset Used	A	A,B,C	B and subset of A, C
Objective	SFT	SFT	SFT+OPD

B Additional Details of the Main Training Run

B.1 Data Mixture

Our training data consists of the following components:

- *A: Text-to-Image Pairs.* Image sources include LAION-2B (Schuhmann et al., 2022), COYO-700M (Byeon et al., 2022), SA-1B (Kirillov et al., 2023), JourneyDB (Sun et al., 2023), BLIP3o-60k (Chen et al., 2025a), ShareGPT4o-Image (Chen et al., 2025b). The images are filtered to remove NSFW prompts, watermarks, low CLIP scores (Radford et al., 2021), low aesthetic scores (Schuhmann, 2022), and low-resolution samples. This results in 127M images in the final mix. Images are recaptioned with Qwen3-VL (Bai et al., 2025a) when only alt-text is available.
- *B: Visual Instruction Tuning Data.* We include LLaVA-OneVision (Li et al., 2024), Open-LLaVA-Next (Chen & Xing, 2024), MAmmoth-VL (Guo et al., 2024), and VisualWebInstruct (Jia et al., 2025). We additionally include grounding-specific data including GranD (Rasheed et al., 2024) and RefCOCO (Kazemzadeh et al., 2014). We additionally include the entire training set of Nemotron-3-Nano-Omni NVIDIA et al. (2026).
- *C: Image Editing Data.* We include ShareGPT4o-Image (Chen et al., 2025b), GPT-Edit-1.5M (Wang et al., 2025b), the image editing subset of UniWorld-V1 (Lin et al., 2025), and ScaleEdit-12M (Chen et al., 2026).

The OPD dataset is sampled from the prompts of the full dataset with labels discarded. Specifically, for visual reasoning OPD, we select 76.8k samples from the datasets in part (B) prioritizing visual reasoning tasks. For image editing tasks, we randomly select 100k samples from datasets in part (C).

B.2 Optimization Setup

We use NLD (Fu et al., 2026b) to initialize the understanding branch and semantic encoder. For the VQ encoder, we adopt a pretrained IBQ (Cui et al., 2025). For the semantic encoder, we adopt RADIO-H (Heinrich et al., 2025). The image generation branch is initialized from the copied weights of the understanding branch. The overall two-tower structure follows Elastic-MoT design proposed by LaViDa-O (Li et al., 2025b).

Training consists of three stages. In the first stage, we fine-tune the language model on image-text pairs from dataset (A). In the second stage, we add instruction data and image editing data from (B) and (C). In the final stage, we use the OPD dataset described above. Since we do not perform OPD on T2I tasks, we mix image-text pairs from (A) to preserve text-to-image ability.

In the main paper, the checkpoint after stage two is referred to as Conflux-Base. The final checkpoint is referred to as Conflux. The full details of the optimization are listed in Table 5.

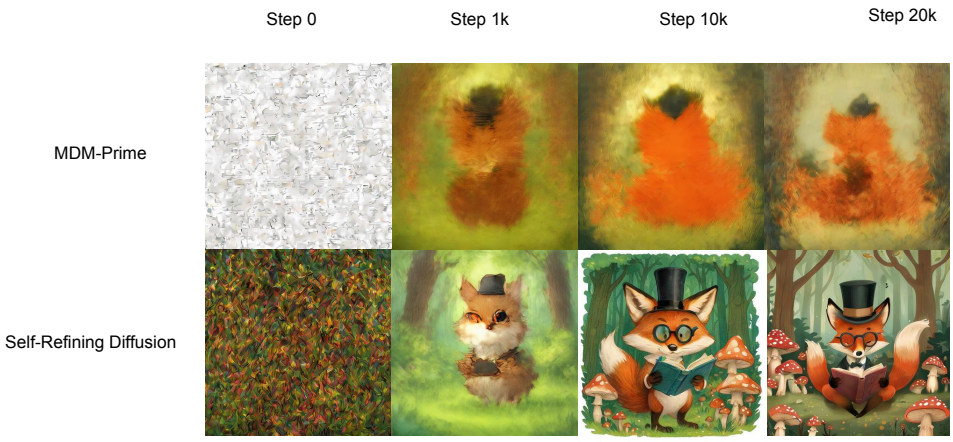


Figure 11: **Comparison with MDM-Prime.** Progression of Generated Images at Different Training Steps. The prompt is *"Highly stylized, whimsical illustration of a fox wearing a tiny monocle and top hat, reading a miniature book in a cozy, mushroom-filled forest glade. Children's book art, gentle colors, charming, adorable."*

B.3 Infrastructure

The training is conducted on 64 H100 GPUs across 8 nodes over 26 days. In particular, the final OPD stage takes 5 days.

C Additional Experiment Results

C.1 Studies on the self-refining diffusion process

C.1.1 Comparison with MDM-Prime

We additionally tried to compare against a related method, MDM-Prime Chao et al. (2025) which also uses partial masking, but it shows no sign of convergence with a score of 0 on GenEval even after 20k steps of training. The progression of generated images throughout training is shown in Figure 11.

Unlike our self-refining process which is shown to correspond to a smooth corruption in pixel space in Figure 8 and Figure 9, MDM-Prime's setup simply factorizes a token index $v \in \mathcal{V}$ into sub-tokens $(v_1, v_2, ..v_l) \in \mathcal{V}_1 \times \mathcal{V}_2 ... \mathcal{V}_l$, with $|\mathcal{V}_i| = |\sqrt[l]{|\mathcal{V}|}|$.

For example, with a vocabulary of 16 tokens and $l = 4$, token 3 is represented as `0011`. The intermediate mask states simply mask the bits, giving states like `0[m][m]1` or `001[m]`. To represent such states, it learns a per-sub-token embedding and concatenates them to form the token embedding. This design materially reduces the total parameters of the embedding table and reduces its expressive power. In MDM-Prime's visual generation setup, it sets $|\mathcal{V}_)| = 16$ which means some dimensions of the token embeddings corresponding to codes $1, 17, 33, ...$ are exactly identical, which can be limiting in large-scale T2I settings because these tokens are not guaranteed to have any similarity at all. We note that MDM-Prime's visual generation experiments are conducted on tiny 32×32 images in CIFAR-10 (Krizhevsky, 2009) and ImageNet-32 (Deng et al., 2009).

We additionally note that while the training failed to converge at 1024×1024 resolution, MDM-Prime's generated images still show some high-level object-like structure, as shown in Figure 11, which could explain why it can succeed on 32×32 low-resolution images.

Table 6: **Effect of Refinement Steps Per Token.** We report HPSv3 scores across actual NFE and number of refinement stages N.

Actual NFE	$N = 1$	$N = 4$	$N = 8$	$N = 16$
4	**7.3700**	-2.4895	—	—
8	**9.6214**	9.0038	0.0834	—
16	10.5449	**10.8205**	10.3044	3.0348
32	10.8921	**11.0137**	10.9894	10.7620
48	10.9378	11.0369	**11.1167**	11.0861
64	11.0051	11.0287	**11.1050**	11.1010

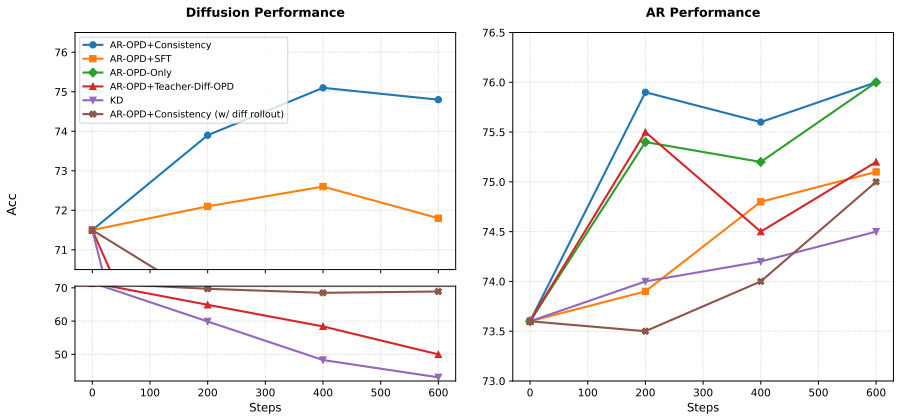


Figure 12: **Ablation Studies on distilling from AR VLMs**. We report MathVista Acc throughout training steps.

C.2 Effect of refinement steps

While in the formulation we use $\epsilon \in [0, 1]$ to define the progression speed of intermediate states in self-refining diffusion, what matters in practical inference settings is the total number of inference steps NFE and the number of refinements each sampled token goes through, denoted by N. For example, $N = 1$ corresponds to vanilla mask diffusion with $\epsilon \to 0$, while $N =$ NFE means all tokens follow the same refinement schedule across the entire sampling process, or $\epsilon = 1$. $N >$ NFE is not possible by definition.

We show the interaction effect of NFE and N in Table 6 at inference time. We note that at any total NFE, setting $\epsilon = 1$ works poorly. At small NFE, $N = 1$ which corresponds to a single state jump works best. By contrast, at larger NFE, a higher number of refinement steps starts to show benefits, with $N = 8$ working best in our default inference setup with 64 steps. Increasing N from 8 to 16 provides no further improvement in the tested settings and can substantially degrade performance at low NFE, where it implies an ϵ value closer to one. This suggests 8 refinement steps are sufficient. The failure patterns at low NFE and around NFE $= N$ suggest we should enforce some form of sequential order between the unmasking/refining processes of different tokens, as opposed to unmasking/refining a large number of tokens simultaneously, which can be challenging.

C.3 Studies on distilling from AR VLMs

We conducted additional ablations on the proposed OPD framework. These results are initialized with Conflux-Base model and post-trained only on multi-modal reasoning tasks. The results are not directly comparable to the main result which reports unified multi-task joint training.

C.3.1 TRAINING OBJECTIVES

We conducted detailed ablations on each component of the proposed OPD framework. Specifically, we compare the following variants of the method:

- (1) AR-OPD+Consistency: The final chosen setup with AR OPD loss and consistency loss.
- (2) AR-OPD+SFT: replace consistency loss with a one-hot SFT objective on the diffusion view, using AR-generated rollouts as targets.
- (3) AR-OPD only: which treats the SBD model as a pure AR model.
- (4) AR-OPD+Teacher-Diff-OPD: directly copy the teacher score as the supervision signal for the diffusion view.
- (5) KD: Replace student's AR rollout with pre-generated teacher diffusion rollout.
- (6) AR-OPD+Consistency (w/ diff rollout): Replace student's AR rollout in (1) with student's diffusion rollout.

We evaluate these settings in both AR generation mode and diffusion generation mode. The results are reported in Figure 12. We draw the following two conclusions:

AR rollouts support effective diffusion supervision. In diffusion generation mode, only (1) and (2) improve upon the baseline because all other setups create mismatched supervision signals caused by mismatched context. In particular, even the student's own diffusion generation fails because the consistency loss uses the student's AR score as the target, which may predict high-probability tokens that are inconsistent given future context generated from the student;s diffusion distribution. This issue is mitigated when the sampled sequences and the supervision target both come from the student's AR distribution.

Consistency Loss does not hurt AR generation. While AR generation is not the focus of this work, we find that the AR performance with AR-OPD+diffusion loss is comparable to AR-OPD-Only, suggesting adding supervision to diffusion view does not negatively affect AR view's performance.

Beyond the above experiments, we also explored a run with consistency loss only. This run quickly collapses to an accuracy of 24.1 at 400 steps, which is outside the plotted range. For presentation clarity, it is not shown in the figure. However, this behavior is not surprising since it ties the outputs of the student's own generation modes without external supervision, which quickly leads to degenerate sequences.

C.3.2 SAMPLING TEMPERATURE

We vary the temperature used to generate student AR rollouts during OPD and evaluate the resulting models in both diffusion and AR decoding modes. As shown in Table 7, greedy rollouts achieve the highest AR accuracy (76.9%), whereas a temperature of 0.4 achieves the highest diffusion accuracy (74.8%). Thus, requiring each rollout token to be an argmax of the student's AR conditional distribution is not necessary for effective transfer to the diffusion view. The improvement at moderate temperatures suggests that greater rollout diversity can benefit diffusion training. Across all tested temperatures, AR-OPD+Consistency achieves higher diffusion accuracy than the alternative designs evaluated in our ablations, such as AR-OPD+SFT. Together, these results support using the student's own AR predictions to supervise its diffusion view, while indicating that strict per-token argmax alignment is unnecessary.

We clarify that all evaluations are performed with a temperature of 0 for consistency and reproducibility. Empirically, even for models with non-zero rollout temperature, using a zero temperature still gives better accuracy.

C.4 STUDIES ON DISTILLING FROM CONTINUOUS FLOW MODELS

We study the effect of noise strength r in the teacher refinement process of image editing OPD with the Qwen-Image teacher. We report benchmark performance in Table 8. Among all tested values, $r = 0.9$ works best. We note that the added Gaussian noise has a highly non-linear effect on the final output. Even with a high noise rate at $r = 0.9$, the generated images are still aligned to the student roll-out and can be very different from a purely teacher-generated sample, as is shown in Figure 10.

Table 7: **Effect of OPD rollout temperature.** We vary the sampling temperature of student AR rollouts during OPD and report MathVista accuracy.

Temperature	Diffusion Performance	AR Performance
0.0	73.9	**76.9**
0.4	**74.8**	76.0
0.6	74.6	75.4
1.0	73.8	75.1

Table 8: **Ablation on Noise Ratio**. Image Editing Results across r values

Method	$\mathbf{r = 0.5}$	$\mathbf{r = 0.8}$	$\mathbf{r = 0.9}$	$\mathbf{r = 1.0}$
ImgEdit	4.06	4.09	4.17	3.90

C.5 Speed-Quality Tradeoff

We examine the speed-quality tradeoff of Conflux by measuring the tokens per forward pass, throughput (tokens per second), and answer accuracy on the MathVista benchmark. Speed-quality trade-off is achieved by tuning the confidence threshold used in the unmasking sampler. Since our model also has an AR view, we report AR's performance. Results are reported in Table 9. In particular, the vanilla setup offers 552.58 tokens/s compared to an AR baseline of 202.74 tokens/s. If we relax the threshold, Conflux can achieve up to 997.95 tokens/s with minor performance degradation.

Table 9: **Speed-Quality Tradeoff on MathVista Benchmark.** Results are measured on H100 GPUs.

Mode / Threshold	Tokens/Forward	Tokens/s/GPU	MathVista (%)
AR	1.00	202.74	**76.1**
Diffusion 0.95	3.94	463.84	75.2
Diffusion 0.90	4.64	552.58	75.0
Diffusion 0.60	**8.31**	**997.95**	73.4
Diffusion 0.50	7.23	734.03	70.8

D Qualitative Results

We present additional qualitative results of text-to-image generation in Figure 13 and image editing in Figure 14. These results highlight the strong generation capability of Conflux.

E Additional Related Works

E.1 Unified Architecture.

Unified MLLMs are capable of both understanding and generation tasks. Integrating different capabilities into a single model presents unique architectural challenges. Generally, unified MLLMs have two common architecture choices. The first type, such as Janus (Chen et al., 2025c), MMaDa (Yang et al., 2025), Show-O(Xie et al., 2024), uses a single dense transformer to output both image and text tokens. The second type, such as BAGEL (Deng et al., 2025), MetaQueries (Pan et al., 2025) and LaViDa-O (Li et al., 2025b) employs separate parameter sets for handling image and text modalities. In particular, mixture-of-transformers (MoT) architecture (Liang et al., 2024) incorporates two separate sets of attention and FFN parameters to process different modalities, while still allowing cross-modal interaction via a joint attention mechanism. Under this paradigm, several works such as X-Fusion and LM-Fusion (Mo et al., 2025; Shi et al., 2024b) further investigated architecture designs and training recipes of MoT. In particular, LaViDa-O (Li et al., 2025b) proposes Elastic-MoT

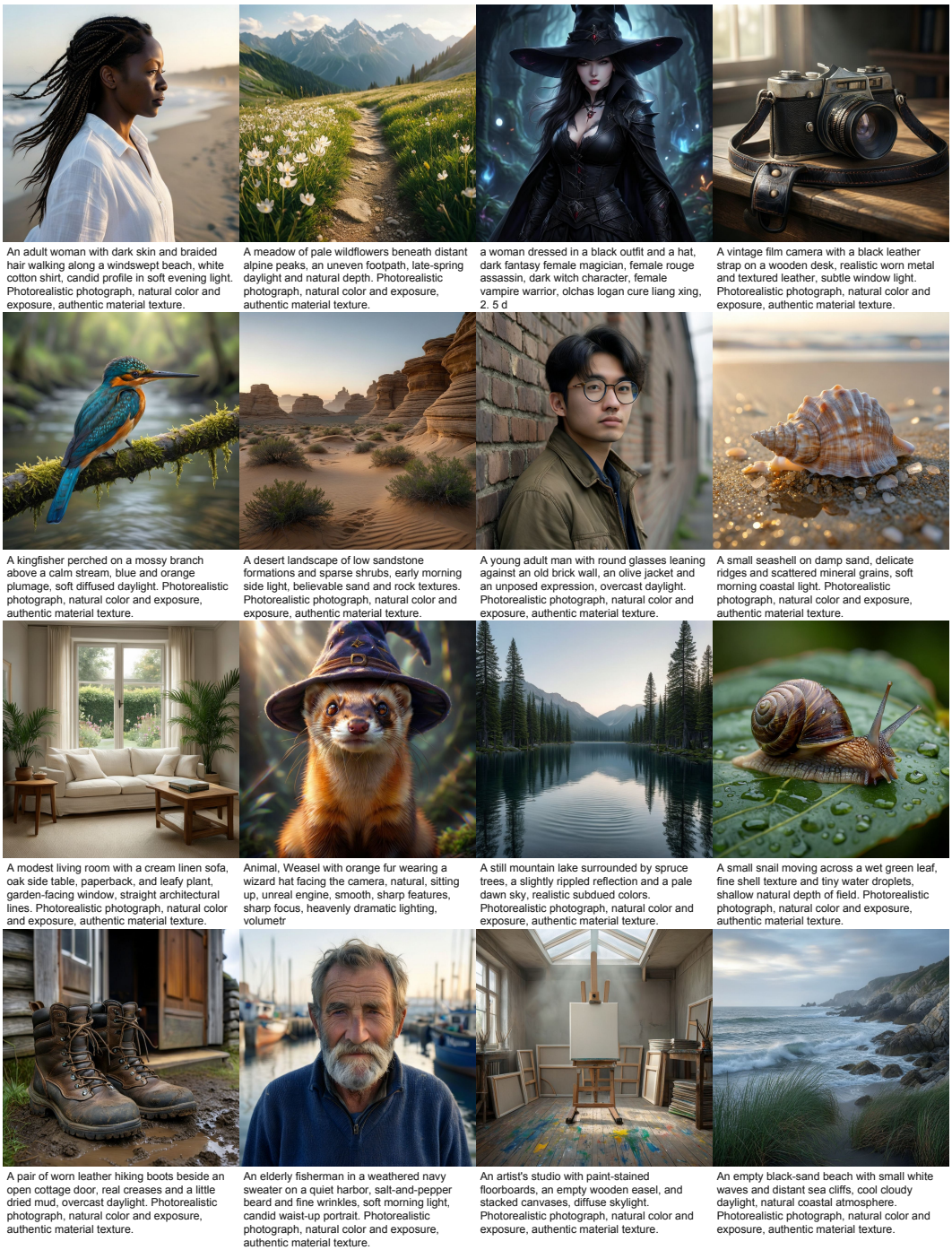


Figure 13: **Qualitative Results on Text-to-Image Tasks**. Results are sampled at 1024×1024 resolution.

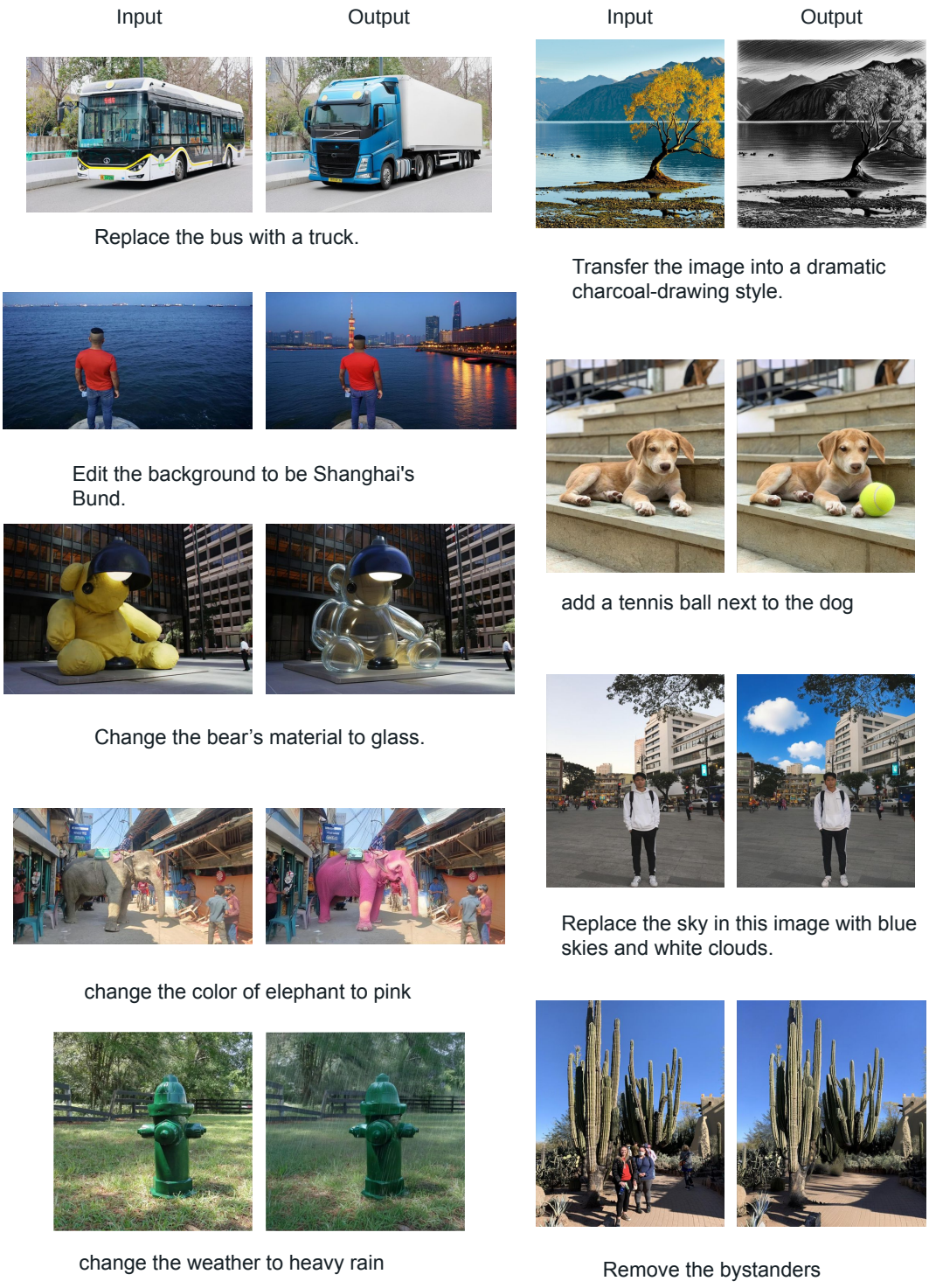


Figure 14: **Qualitative Results on Image Editing Tasks**. For each input-output pair, we present the input image on the left and the output image on the right.

design, which decouples the attention layer in the final half of the model and only enables joint attention in early layers. Given an 8B base model, Elastic-MoT extends it to a 16B MoT. However, when performing understanding tasks, only 8B parameters are loaded. When performing generation tasks, only 8B generation parameters + 4B understanding parameters are loaded. When performing interleaved tasks, all 16B parameters are loaded. This elasticity makes the model particularly efficient and introduces no overhead compared to an 8B VLM when performing understanding tasks.

F List of Benchmarks

.

In this section, we provide a full list of benchmark used in this work and their source references.

- **Generic Multi-modal Understanding**: MMMU (Yue et al., 2024),RealWorldQA (xAI, 2024), ChartQA (Masry et al., 2022), DocVQA (Mathew et al., 2021), InfoVQA Mathew et al. (2022), TextVQA (Singh et al., 2019), AI2D (Kembhavi et al., 2016), RefCOCO (Kazemzadeh et al., 2014), ScienceQA (Lu et al., 2022b), ScreenSpot-Pro (Li et al., 2025a), OCRBench-V2 (Fu et al., 2026a).
- **Multi-modal Reasoning**: (Lu et al., 2023), (Zhang et al., 2024), (Wang et al., 2024a), DynaMath (Zou et al., 2025), Geometry3k (Lu et al., 2021), LogicVista (Xiao et al., 2024), We-Math (Qiao et al., 2025).
- **Multi-modal Generation**: GenEval (Ghosh et al., 2023), DPG (Hu et al., 2024), HPSv3 Ma et al. (2025), ImgEdit Ye et al. (2025).

G Limitations and Future Work

In this work, we propose Conflux, introducing a self-refining diffusion process and cross-paradigm on-policy distillation. Despite considerable success, some limitations remain. First, while the cross-paradigm OPD framework allows us to distill an AR teacher to a diffusion student, it still requires the vocabulary spaces of the two models to match. In our setup, both models use the Ministral 3 (Liu et al., 2026a) tokenizer. Many existing works have explored cross-tokenizer distillation between AR models Niu et al. (2026). We leave extending these methods to support cross-paradigm distillation to future work. Second, while Conflux achieves state-of-the-art performance among unified diffusion MLLMs, there may be considerable room for further optimization on the model architecture side. We leave that for future work as well since it is not the focus of this work. Third, we define the self-refining diffusion process with the non-Markov formulation with predefined per-token corruption intervals and uniform progression of intermediate states. This design is mostly motivated by minimizing complexity. Future work may explore defining a Markov process with a stochastic transition rate that allows tokens to stay at different intermediate states for varying lengths of time. Finally, from a system perspective, Conflux lacks fine-grained text rendering capabilities because such data is not widely present in the training mix. We will address this issue in future by curating additional text-rendering-focused data.