Sparse-LaViDa: Sparse Multimodal Discrete Diffusion Language Models

Shufan Li[1,2,*], Jiuxiang Gu[1], Kangning Liu[1], Zhe Lin[1]
Zijun Wei[1], Aditya Grover[2], Jason Kuen[1]
[1]Adobe [2]UCLA

Abstract

Masked Discrete Diffusion Models (MDMs) have achieved strong performance across a wide range of multimodal tasks, including image understanding, generation, and editing. However, their inference speed remains suboptimal due to the need to repeatedly process redundant masked tokens at every sampling step. In this work, we propose Sparse-LaViDa, a novel modeling framework that dynamically truncates unnecessary masked tokens at each inference step to accelerate MDM sampling. To preserve generation quality, we introduce specialized register tokens that serve as compact representations for the truncated tokens. Furthermore, to ensure consistency between training and inference, we design a specialized attention mask that faithfully matches the truncated sampling procedure during training. Built upon the state-of-the-art unified MDM LaViDa-O, Sparse-LaViDa achieves up to a 2× speedup across diverse tasks including text-to-image generation, image editing, and mathematical reasoning, while maintaining generation quality.

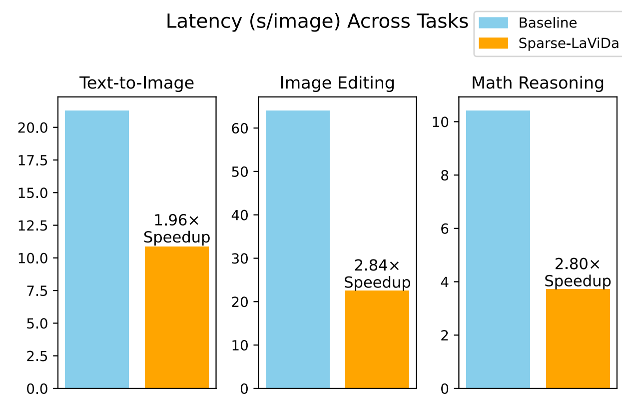


Figure 1. **We propose Sparse-LaViDa**, a novel modeling technique for unified multimodal masked discrete diffusion models. Sparse-LaViDa achieves substantial speedup across a wide range of tasks, including text-to-image generation, image editing, and visual math reasoning, compared with the baseline LaViDa-O.

1. Introduction

Improving visual understanding and generation capabilities has been a major focus of artificial intelligence research. A common paradigm is to employ autoregressive (AR) vision-language models (VLMs) for visual understanding tasks such as question answering, and diffusion models for visual generation tasks such as text-to-image generation and image editing. Recently, there has been rising interest in building unified multimodal models capable of both understanding and generation [12, 40, 63, 76]. These models often surpass the performance of single-task systems because they enable understanding and generation capabilities to mutually benefit from shared representations under a unified framework, which is especially advantageous for tasks requiring both abilities, such as image editing.

Early works such as Transfusion [76] and BAGEL [12] build unified multimodal models by combining AR VLMs with continuous diffusion models to handle understanding and generation tasks respectively. More recently, Masked Diffusion Models (MDMs) have emerged as a promising alternative, offering a unified modeling framework for both text and image generation [32, 33, 54, 66, 71]. Concretely, MDMs represent both text and images as sequences of discrete tokens. Given such a sequence, the forward diffusion process gradually converts it into a fully masked sequence. An MDM learns the reverse process by predicting the distribution of original tokens at the masked positions. To sample from MDMs, we start with an all-mask sequence and iteratively unmask tokens to obtain a clean sequence. This formulation brings several advantages over AR models, such as faster inference via parallel decoding, controllable generation, and bidirectional context, *etc*. (Figure 2 Left) Notably, the unified MDM LaViDa-O [32] achieves strong performance across a wide range of image understanding and generation tasks.

Despite supporting parallel decoding, existing MDMs still face major efficiency limitations. First, they rely on full attention instead of the causal attention used by AR models. While this enables bidirectional context and nat-

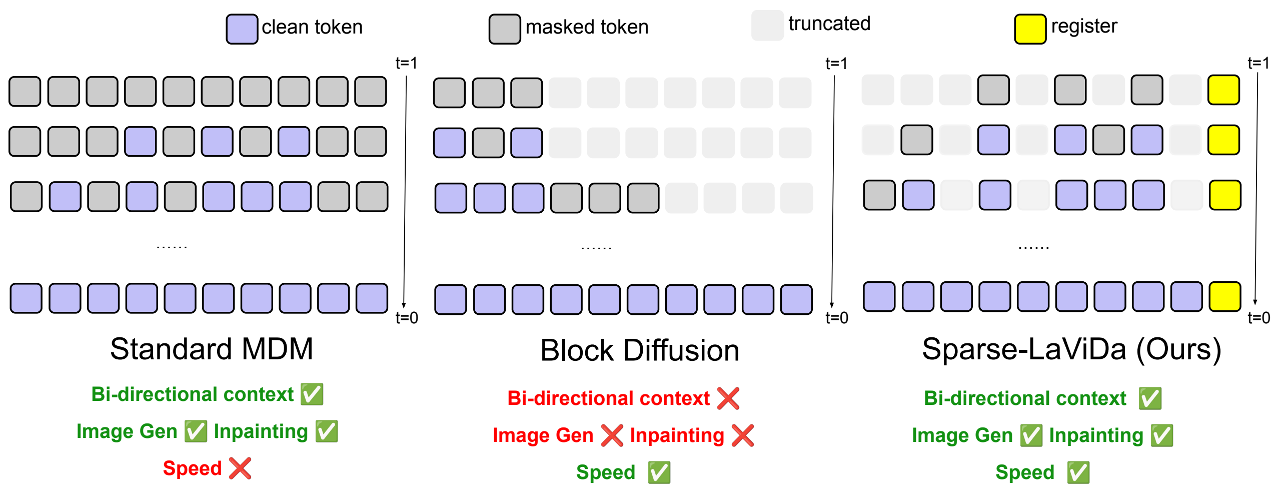


Figure 2. **Overall design of Sparse-LaViDa.** *Left:* Vanilla MDMs materialize all masked tokens and support arbitrary-order decoding (top-down). Unlike AR models, they have bidirectional context and naturally support tasks such as image generation and inpainting. *Middle:* Block Diffusion truncates redundant masked tokens from the right but imposes a left-to-right generation order using a block-causal attention mask, losing many benefits of MDMs. *Right:* Sparse-LaViDa is an alternative parameterization of vanilla MDMs. It preserves all benefits of standard MDMs while achieving the efficiency gains of Block Diffusion by allowing mask truncation at arbitrary positions. Special register tokens serve as compressed representations of truncated tokens.

urally supports tasks such as text infilling and image inpainting, it prevents the use of KV-cache acceleration during inference. Second, they must repeatedly process the full sequence, including many redundant masked tokens, at every sampling step. For example, if an image is represented by 1024 tokens, the model process all 1024 tokens at each diffusion step, even though only a small subset is unmasked at a time. Recently, Block Diffusion [1] was proposed to improve MDM efficiency by constraining parallel decoding to a left-to-right block-causal order (Figure 2, Middle). While effective for language modeling, this left-to-right generation scheme is poorly suited for image generation and editing tasks, where tokens do not follow a natural ordering. Furthermore, its block-causal attention mask eliminates bidirectional context,a key advantage of MDMs, making tasks such as inpainting difficult.

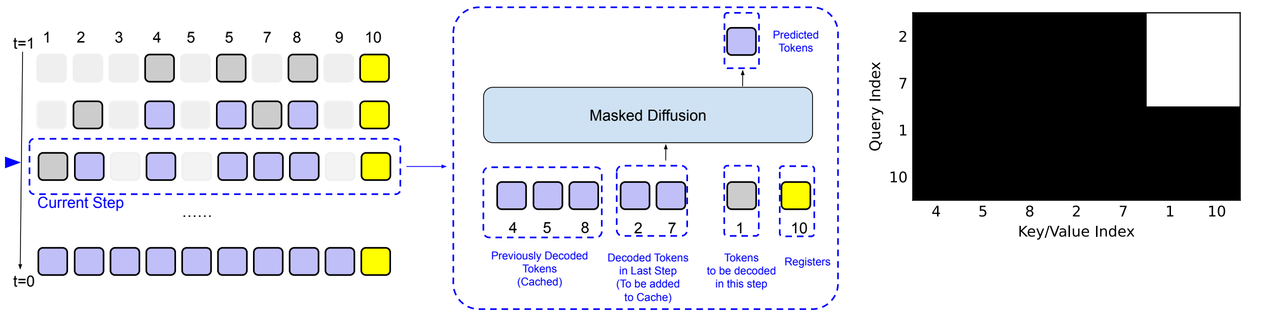
To address these limitations, we propose Sparse-LaViDa, a novel modeling framework that improves the efficiency of MDMs by supporting KV-cache usage and enabling truncation of arbitrary subsets of redundant tokens during inference. Concretely, Sparse-LaViDa introduces three key innovations. First, we propose a sparse parameterization of MDMs that represents partially masked sequences without materializing all masked tokens. At each sampling step, Sparse-LaViDa takes only the prompt tokens, previously generated tokens, and a selected subset of masked tokens that need to be decoded, in contrast to vanilla MDMs which always materialize all masks. Second, we introduce special register tokens that serve as compressed representations of truncated tokens and help recover modeling capacity lost due to truncation. Finally, we design a step-causal attention mask that enables KV-cache support during inference while allowing efficient training. Unlike the block-causal mask used in Block Diffusion, our step-causal attention mask preserves the bidirectional context essential for image generation, editing, and inpainting. These components are illustrated in Figure 2 (Right).

To validate the effectiveness of Sparse-LaViDa, we conduct extensive experiments across image generation, understanding, and editing tasks. Sparse-LaViDa achieves significant efficiency gains, including a $1.96\times$ speedup on text-to-image generation, $2.80\times$ speedup on image editing, and $2.84\times$ speedup on visual math reasoning, while maintaining generation quality comparable to the unified MDM LaViDa-O [32] (Figure 1). Notably, these improvements are achieved on top of many optimizations already employed by LaViDa-O, such as token compression.

2. Related Work

2.1. Masked Diffusion Models

Early works on masked modeling, such as BERT [13] and MAE [19], focus on learning semantically rich representations through masked autoencoding. MaskGIT [7] and Muse [8] were among the first to apply masked modeling to image generation, using VQGAN [14] to convert images into sequences of discrete tokens. Despite some success, these early approaches lacked a strong theoretical foundation and relied heavily on heuristics. Recently, MDMs [2, 38, 51] have established a principled theoretical



(a) Inference with Sparse Parameterization

(b) Inference Attention Mask of Sparse-LaViDa

Figure 3. **Inference pipeline of Sparse-LaViDa.** (a) At a specific decoding step (top-down), the input of Sparse-LaViDa consists of four types of tokens: (1) previously decoded tokens stored in the KV cache; (2) newly decoded tokens from the previous step, which will be added to the cache at this step; (3) masked tokens to be decoded at the current step; and (4) register tokens. (b) During sampling, we apply a specialized attention mask such that tokens of type (2) cannot attend to tokens of type (3) or (4). In this example, tokens at positions 2 and 7 are of type (2) (newly decoded), while tokens at positions 1 and 10 correspond to types (3) and (4) (mask and register), respectively.

framework by formulating the masking and unmasking processes as forward and reverse discrete diffusion processes. This provides a unified and mathematically grounded approach for training and sampling in masked generative models. Building on this foundation, several works such as Mercury [24], LLaDA [45] and Dream [67] have successfully scaled MDMs to large-scale language modeling, achieving performance comparable to autoregressive counterparts while offering advantages such as bidirectional context and parallel decoding. Further extensions, including LaViDa, LaViDa-O, and MMaDa [32, 33, 54, 66, 71], have expanded MDMs to multimodal understanding and generation tasks, achieving impressive results.

Formally, given a sequence of discrete tokens $X_0 = [X_0^1, X_0^2, \ldots, X_0^L]$, where L denotes the sequence length, the forward masked diffusion process $q(X_t|X_s)$ progressively replaces clean tokens with masked tokens over the time interval $[0, 1]$, with $1 \geq t \geq s \geq 0$. At $t = 1$, the sequence $X_1 = [M, M, \ldots, M]$ consists entirely of masked tokens. For intermediate steps where $0 < t < 1$, X_t contains a mixture of clean and masked tokens. A neural network p_θ is trained to model the reverse process $p(X_s|X_t)$. The masked diffusion objective is defined as:

$$\mathcal{L}_{\text{MDM}} = -\mathbb{E}_{t,X_0,X_t}\left[\frac{1}{t}\log p_\theta(X_0|X_t)\right], \tag{1}$$

where $p_\theta(X_0|X_t)$ is factorized as $\prod_{i=1}^{L} p_\theta(X_0^i|X_t)$ under standard independence assumptions [51]. At inference, we begin with a fully masked sequence X_1 and iteratively apply the learned reverse process $\log p_\theta(X_0|X_t)$ to progressively unmask tokens until a clean sequence X_0 is obtained.

Most existing MDMs adopt a dense parameterization. At intermediate steps where $0 < t < 1$, all L tokens in X_t, both masked and unmasked, are passed to the neural network p_θ, which outputs a dense tensor $y \in \mathbb{R}^{L\times V}$, where V is the vocabulary size. Each $y[i] \in \mathbb{R}^V$ represents the logits corresponding to $\log p_\theta(X_0^i|X_t)$. This design is computationally inefficient because all L tokens must be processed even when predictions are only needed for a small subset. The key contribution of Sparse-LaViDa is a sparse parameterization that directly addresses this inefficiency.

2.2. Acceleration Methods for MDMs

MDMs are known to suffer from inefficiencies since they do not natively support KV caching. Several approaches, such as Fast-dLLM [60], dKV-Cache [41], and Sparse-dLLM [55], attempt to incorporate KV caching into MDMs in a training-free manner through heuristic modifications. However, these methods mostly focus on diffusion large language models (dLLMs) and assume a left-to-right, block-wise, semi-AR sampling scheme. Moreover, while training-free, these approaches often result in unpredictable performance degradation that varies across tasks.

Training-based acceleration methods have also been proposed for MDMs. Prominent examples include Block Diffusion [1], SDAR [11], and D2F [57], which interpolate between autoregressive and diffusion modeling by introducing block-causal attention masks. Compared with heuristic, training-free methods, these approaches natively support KV caching without inference-time performance degradation and achieve greater speedups by truncating redundant tokens. However, similar to the previous category, they are mostly designed for language modeling and assume a left-

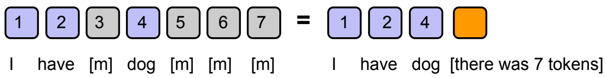


Figure 4. **Illustration of Sparse Representation of Masked Sequence.** Instead of materializing all masked tokens, a partially masked sequence can be uniquely represented by non-mask tokens, their locations, and the number of total tokens in the original sequence.

to-right semi-AR decoding order. Furthermore, they sacrifice the bidirectional context of MDMs, which is crucial for tasks such as image generation and inpainting.

Sparse-LaViDa is the first method to support both KV caching and token truncation without assuming a left-to-right decoding order or sacrificing bidirectional context. It implements the standard MDM formulation described in Sec. 2.1 efficiently and faithfully, preserving all desirable properties of MDMs without any qualitycompromise or training–inference gap.

3. Method

3.1. Sparse Parameterization of Masked Sequence

The core idea of Sparse-LaViDa is based on the independence of assumption described in Sec. 2.1, where $p_\theta(X_0|X_t)$ is factorized into $\prod_{i=1}^{L} p_\theta(X_0^i|X_t)$. Since $p_\theta(X_0^i|X_t)$ are optimized in the training and sampled at inference independently, if we are only interested in the model's prediction at a subset of locations C, there is no reason to predict $p_\theta(X_0^j|X_t)$ where $j \notin C$. However, this observation alone does not allow us to truncate any input tokens, since X_t^j is still part of X_t which are needed to compute $p_\theta(X_0^j|X_t)$.

Recall that X_t is a partially masked sequence containing both clean tokens and mask tokens. Critically, masked tokens carry no substantive information beyond indicating that a position was masked, making compression possible. Consider a seven-token sequence "I have [m] dog [m] [m] [m]". Rather than representing this with 7 independent tokens, we can equivalently encode it using: (1) clean tokens with their positional embeddings ("I" at position 1, "have" at position 2, "dog" at position 4), and (2) a special token indicating the total sequence length (e.g., 7). The positions of masked tokens are then implicitly determined, as they occupy all positions not taken by clean tokens. In other words, specifying the sequence length is sufficient to represent which positions are masked.

Register Tokens. In practice, we find that using a single special token is not sufficient and leads to considerable performance drop on image generation quality. There are two reasons. First, truncating tokens may leads to considerable drop in model capacity. For example, an 1024×1024 image is represented by 4096 tokens. In the first few sampling steps, we only sample less than 100 tokens. Although aggressively reducing the token count improves efficiency, it does so at the expense of the model's capacity. Second, previous works[29] using special tokens for text-to-image generation show that a sufficient number of special tokens is needed to meaningfully impact the generation process through attention mechanism. In Sparse-LaViDa , we use 64 special register tokens in our final design, whose position ids are consecutively located at the end of the sequence (in the above example, it will be 8-51). This number remains constant throughout the inference process and is small in relation to the total sequence length. It does not grow with the number of truncated masked tokens.

3.2. Sampling with Sparse Parameterization

Given a prompt p, we sample a sequence of response tokens X_0 (image or text) starting from a fully masked sequence X_1. We first prefill the KV cache with prompt tokens from p. At any sampling step k, the model input consists of the prompt p, all previously decoded tokens $(C_1 \dots C_{k-1})$, and the tokens to be decoded (C_k). Of these inputs, p and $C_1 \dots C_{k-2}$ are already in the KV cache. The **new cache tokens** C_{k-1} (decoded in the previous step) are processed and added to the cache in this step. Then, the model produces logits only for the **decode tokens** C_k, which are sampled to unmask them. This process repeats until all tokens are unmasked.

To enable proper KV cache updates, we apply a specialized attention mask. Queries from C_{k-1} attend only to $\{p, C_1, \dots, C_{k-1}\}$ and cannot attend to C_k. This design ensures cached representations are unaffected by masked tokens, enabling efficient training (see Sec. 3.3). We also include register tokens R at each step, which can attend to all tokens but are only attended to by C_k and R. This process is illustrated in Figure 3.

The remaining question to be addressed is how to decide the order of unmasking $C_1 ... C_N$ where N is the number of sampling steps. This strategy differs according to the task.

Text-to-Image and Image Editing. For visual generation tasks, several works [4, 32] have proposed to use pre-generated 2D sequences as the unmasking order. Compared with confidence-based approaches which dynamically decides which token to unmask at each step based on confidence score, these pre-generated unmasking order have been shown to have higher generation quality. We use the stratified random sampler proposed by LaViDa-O [32], which generates an unmasking order without the need for confidence scores. This allows us to easily set up $C_1 ... C_N$ based on a pre-generated unmasking order.

Text generation and Image understanding. Unlike image generation and editing tasks, language generation of MDMs typically employ semi-auto-regressive sampling strategy. A sequence of length L is first divided into blocks of equal size S. These $\frac{L}{S}$ blocks are sampled auto-regressively from left-to-right. When sampling tokens with in a block, we pass in all tokens in the same block and dynamically decide which tokens to unmask based on confidence scores. In this setup, while we do not know the exact tokens that will be unmasked at k-th step C_k, we know it belongs to some block $B \supset C_k$. In this step, the input to the model at kth step includes the prompt p, all previously decoded tokens $C_1 ... C_{k-1}$ and all tokens in B. $C_k \subset B$ is determined after the forward pass based on per-token confidences. While the default left-to-right behavior is similar to Block Diffusion, we highlight that Sparse-LaViDa still supports bi-directional context and arbitrary decoding order of blocks, without assuming a left-to-right block-causal attention masks. This means it can uniquely perform tasks such as text-infilling, constraint generation while methods like block diffusion are not capable because they only have one-sided context. Some of these examples are shown in Sec. 4.5.

3.3. Training Pipeline

Step-Causal Attention Mask. To enable efficient parallel training while maintaining consistency with the inference of Sparse-LaViDa, we design a step-causal attention mask that simulates the incremental token caching behavior observed during inference. In a vanilla MDM training step, we have a prompt p and a partially masked response X_t, and the model learns to predict the clean sequence X_0 given (p, X_t). Full attention is applied, and all tokens may interact freely. However, this is incompatible with Sparse-LaViDa, since KV caching and token truncation imply that during inference: (1) not all tokens can attend to each other, and (2) the model may not observe all tokens.

To close this training–inference gap, we partition the sequence X_t into $M + N$ blocks and apply a structured attention mask. Prompt tokens are assigned block number 0. Clean tokens in X_t are randomly assigned block numbers in $\{1, \ldots, M\}$. Masked tokens are randomly assigned block numbers in $\{M + 1, \ldots, M + N\}$.

A clean token in block $i \in \{1, \ldots, M\}$ may only attend to tokens in blocks $j \leq i$. For masked tokens in block $i \in \{M + 1, \ldots, M + N\}$, attention is allowed only to blocks $j = i$ (same masked block) or $j \leq M$ (prompt and clean tokens). Thus, masked tokens may interact with prompt and clean tokens, but not with masked tokens in other blocks.

For every masked block, we append corresponding register tokens to the sequence and assign them the same block number. They follow the same attention rules as masked tokens. We note that although we use the term "block," tokens with the same block number need not form a contiguous segment in X_t. This design is illustrated in Fig. 5.

Training Objective. Since Sparse-LaViDa is an alternative parameterization of the standard MDM, it uses the standard MDM training objective. At each step, we sample

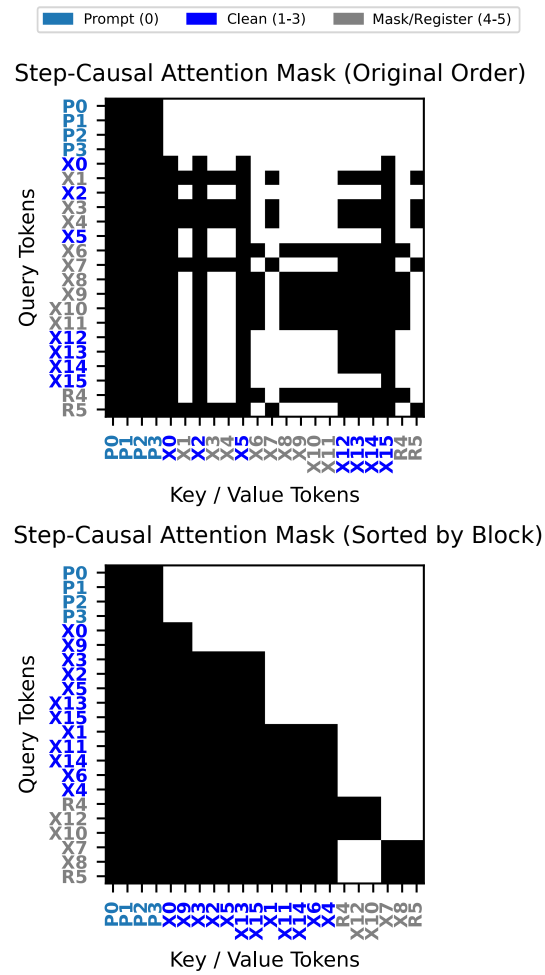


Figure 5. **Step-Causal Mask.** We employ a step-causal attention mask during training to match the inference behavior of Sparse-LaViDa. Consider a sequence containing prompt tokens P_0–P_3 and answer tokens X_0–X_{15}, where some tokens are clean and others are masked (color-coded in blue and gray). During inference, prompt tokens and clean tokens are sequentially added to the KV cache. To simulate this behavior during training, we assign block number 0 to the prompt, and block numbers 1–3 to clean tokens, such that each token may only attend to tokens in its own block or previous blocks. The bottom figure shows tokens sorted by block number.At inference, the model only observes a subset of masked tokens. To mimic this behavior in training, we assign block numbers 4–5 to masked tokens (e.g., X_{10}, X_{12} in block 4 and X_7, X_8 in block 5). Each block is accompanied by a corresponding register token (R_4, R_5). We apply an attention mask such that tokens in one masked block cannot attend to tokens in another masked block, but may attend to all clean and prompt tokens. This simulates inference paths $0 \to 1 \to 2 \to 3 \to 4$ and $0 \to 1 \to 2 \to 3 \to 5$ within a single training step.

Table 1. **Text to Image Generation Performance on GenEval Dataset**. *These models do not support 1024px generation.

	Parms	Single↑	Two ↑	Position↑	Counting↑	Color↑	Attribution↑	Overall↑	Latency↓	Speedup↑
SDXL[48]	2.6B	0.98	0.74	0.39	0.85	0.15	0.23	0.55	5.2	-
DALLE 3[46]	-	0.96	0.87	0.47	0.83	0.43	0.45	0.67	-	-
SD3[15]	8B	0.99	0.94	0.72	0.89	0.33	0.60	0.74	23.3	-
Flux-Dev[27]	12B	0.99	0.85	0.74	0.79	0.21	0.48	0.68	31.6	-
Playground v3[30]	-	0.99	0.95	0.72	0.82	0.50	0.54	0.76	-	-
BAGEL [12]	14B	0.99	0.94	0.64	0.81	0.88	0.63	0.82	45.1	-
Show-o [65]	1B	0.98	0.80	0.31	0.66	0.84	0.50	0.68	*	-
MMaDa[66]	8B	0.99	0.76	0.20	0.61	0.84	0.37	0.63	*	-
LaViDa-O [32]	10.4B	0.99	0.85	0.65	0.71	0.86	0.58	0.77	21.27	1.00 ×
Sparse-LaViDa	10.4B	0.99	0.93	0.63	0.61	0.88	0.64	0.78	10.86	1.95 ×

a ground truth sequence X_0 from the data distribution (containing both images and text), sample a time $t \in [0, 1]$, and draw $X_t \sim q(X_t \mid X_0)$ from the forward masking process. We then optimize the MDM objective Eq. (1) on the model output $p_\theta(X_0 \mid X_t)$.

The only difference from vanilla MDM training lies in how we implement $p_\theta(X_0^i \mid X_t)$, following our sparse parameterization and step-causal masking. Unlike methods such as D2F [57], which rely on distillation-based objectives for post-hoc acceleration, Sparse-LaViDa provides a fundamentally efficient parameterization that supports scalable training and inference without additional distillation stages.

4. Experiments

4.1. Setup

We initialize Sparse-LaViDa with the pretrained weights of LaViDa-O [32], a state-of-the-art 10.4B unified diffusion model that supports a wide range of multimodal tasks, including image understanding, text-to-image generation, and image editing.

Training. We perform supervised fine-tuning (SFT) on a mixture of image understanding, generation, and editing datasets to adapt LaViDa-O's dense parameterization to the sparse design of Sparse-LaViDa. We source image understanding data from MAmmoth-VL [18] and VisualWebInstruct [22]. For text-to-image generation, we subsample 20M text–image pairs from LAION-2B [53], COYO-700M [6], SA-1B [25], JourneyDB [56], BLIP3o-60k [10], and ShareGPT4o-Image [9]. We include GPT-Edit-1.5M [59] for image editing. Overall, our training dataset is a filtered subset of the LaViDa-O SFT data, selected for higher quality and efficient fine-tuning. We train for 100k steps on 64 NVIDIA H100 GPUs. Details of our data pipeline and hyperparameters are provided in the Appendix.

Evaluation. We conduct extensive evaluations across diverse multimodal benchmarks to demonstrate the effectiveness of Sparse-LaViDa, including GenEval [17], DPG [21], and MJHQ-30k [30] for text-to-image generation; ImgEdit [68] for image editing; and a suite of image understanding benchmarks [16, 37, 39, 43, 44, 72, 74]. We also report inference latency (seconds per image) and relative speedup with respect to the base model LaViDa-O. Unless otherwise stated, all image generation experiments are performed at 1024 resolution on a single A100 GPU.

Table 2. **Text-to-Image Generation Results on DPG-Bench and MJHQ-30K.** We report the benchmark score of DPG and PickScore, HPS v3, HPS v3, and FID on MJHQ-30k. *We perform SFT on the same data mix as Sparse-LaViDa.

	DPG↑	MJHQ-30k↑			
		PickScore↑	HPS v2↑	HPS v3 ↑	FID↓
LaViDa-O [32]	81.8	21.02	0.271	8.81	6.68
LaViDa-O* [32]	82.1	21.04	0.297	8.87	8.11
Sparse-LaViDa	82.4	21.04	0.291	8.89	7.63

4.2. Text-to-Image Generation

We report text-to-image generation results on the GenEval benchmark [17] in Table Tab. 1. We compare against the base model LaViDa-O and other state-of-the-art text-to-image models such as Flux.1-Dev [27] and unified multimodal models such as MMaDa [66]. Sparse-LaViDa achieves performance comparable to LaViDa-O (+0.01) while substantially reducing end-to-end latency (21.27s vs. 10.86s), achieving a 1.95× speedup. It also surpasses models such as Flux.1-Dev in both performance and efficiency, highlighting the effectiveness of our sparse parameterization.

To further assess performance, we evaluate Sparse-LaViDa on DPG-bench [21] and MJHQ-30k [30]. DPG-bench is evaluated using a VQA model for prompt alignment, while MJHQ-30k reports FID and reward-based

Table 3. **Image Editing Performance on ImgEdit benchmark.** We report per-category scores and the overall scores.

Model	Add ↑	Adjust ↑	Extract↑	Replace↑	Remove↑	Background↑	Style↑	Hybrid↑	Action ↑	Overall↑	Latency↓	Speedup↑
GPT-4o [47]	4.61	4.33	2.90	4.35	3.66	4.57	4.93	3.96	4.89	4.20	111.4	-
Qwen2.5VL+Flux [59]	4.07	3.79	2.04	4.13	3.89	3.90	4.84	3.04	4.52	3.80	55.2	-
FluxKontext dev [28]	3.76	3.45	2.15	3.98	2.94	3.78	4.38	2.96	4.26	3.52	51.4	-
OmniGen2 [61]	3.57	3.06	1.77	3.74	3.20	3.57	4.81	2.52	4.68	3.44	84.8	-
UniWorld-V1 [34]	3.82	3.64	2.27	3.47	3.24	2.99	4.21	2.96	2.74	3.26	56.2	-
BAGEL [12]	3.56	3.31	1.70	3.30	2.62	3.24	4.49	2.38	4.17	3.20	88.2	-
Step1X-Edit [36]	3.88	3.14	1.76	3.40	2.41	3.16	4.63	2.64	2.52	3.06	-	-
OmniGen [64]	3.47	3.04	1.71	2.94	2.43	3.21	4.19	2.24	3.38	2.96	126.2	-
UltraEdit [75]	3.44	2.81	2.13	2.96	1.45	2.83	3.76	1.91	2.98	2.70	-	-
AnyEdit [70]	3.18	2.95	1.88	2.47	2.23	2.24	2.85	1.56	2.65	2.45	-	-
InstructAny2Pix[31]	2.55	1.83	2.10	2.54	1.17	2.01	3.51	1.42	1.98	2.12	48.2	-
MagicBrush [73]	2.84	1.58	1.51	1.97	1.58	1.75	2.38	1.62	1.22	1.90	-	-
Instruct-Pix2Pix[5]	2.45	1.83	1.44	2.01	1.50	1.44	3.55	1.20	1.46	1.88	9.5	-
LaViDa-O [32]	4.04	3.62	2.01	4.39	3.98	4.06	4.82	2.94	3.54	3.71	63.98	1.00×
Sparse-LaViDa	4.08	3.73	2.10	4.29	3.98	4.06	4.84	3.30	3.76	3.79	22.55	2.83×

metrics including PickScore [26], HPS v2 [62], and the latest VLM-based reward HPS v3 [42] (higher is better). As shown in Tab. 2, Sparse-LaViDa outperforms the LaViDa-O baseline on DPG-bench (+0.6). On MJHQ-30K, Sparse-LaViDa achieves superior results across all perceptual metrics except FID (lower is better), which increases marginally by less than one point. Importantly, when both models are trained on the same 20M subset, our Sparse-MDM (FID 7.63) outperforms the baseline LaViDa-O* (FID 8.11), demonstrating that the sparse parameterization and step-causal training not only maintain but can improve generation quality under identical data conditions.

4.3. Image Editing

We evaluate image editing performance on the ImgEdit benchmark [68], which measures both visual quality and prompt compliance via a GPT-4 judge model. Sparse-LaViDa achieves higher accuracy (+0.08) compared to LaViDa-O and other state-of-the-art unified models such as BAGEL [12], while reducing end-to-end latency from 63.98s to 22.55s, achieving a 2.83× speedup.

4.4. Image Understanding

To assess text generation efficiency, we evaluate Sparse-LaViDa on the MathVista reasoning benchmark with generation length set to $L = 1024$ tokens and block size $S = 32$. We compare against LaViDa-O's vanilla sampling and Fast-dLLM [60], a training-free KV-caching baseline. Results in Table Tab. 4 show that Sparse-LaViDa matches the base model's accuracy while achieving a 2.80× speedup. It also outperforms Fast-dLLM in both accuracy and latency, confirming the advantage of our learned truncation strategy.

For completeness, we also report Sparse-LaViDa on additional understanding benchmarks including MME-C [16], MMMU [72], ChartQA [43], DocVQA [44], and MathVerse [74]. These results (Table 5) show that Sparse-LaViDa achieves competitive performance across all benchmarks. However, speedups are minimal for short QA tasks where outputs contain fewer tokens than one block (32 tokens), effectively reducing Sparse-LaViDa to prompt caching without truncation.

Table 4. **Quantative Results on Visual Math Reasoning.** We compare Sparse-LaViDa with other caching strategies on MathVista accuracy and latency.

Model	MathVista↑	Latency↓	Speedup↑
LaViDa-O [32]	56.9	10.41s	1.00 ×
LaViDa-O+Fast-dLLM [60]	56.1	5.57s	1.87 ×
Sparse-LaViDa	56.7	3.72s	2.80 ×

Table 5. **Image Understanding Performance.** We report performance on a wide range of image understanding tasks and compare the performance of Sparse-LaViDawith LaViDa-O baseline.

Model	MME	MMMU	MMB	ChartQA	DocVQA	MathVista	MathVerse
LaViDa-O [32]	488	45.1	76.4	80.0	73.7	56.9	36.9
Sparse-LaViDa	450	43.6	75.0	82.0	75.7	56.7	37.9

4.5. Qualitative Results

In Fig. 6, we present qualitative examples across understanding and generation tasks, including text-to-image generation and image editing. Notably, unlike semi-autoregressive methods such as Block Diffusion, Sparse-LaViDa natively supports tasks requiring bidirectional context, such as image inpainting/outpainting, parallel object grounding, and constrained captioning.

4.6. Ablation Studies

Effect of token caching and truncation. The speed advantage of Sparse-LaViDa primarily arises from two sources: token caching and truncation. To isolate their contributions, we perform ablations on text-to-image (T2I) tasks. Specifically, we decompose speedup into three components:

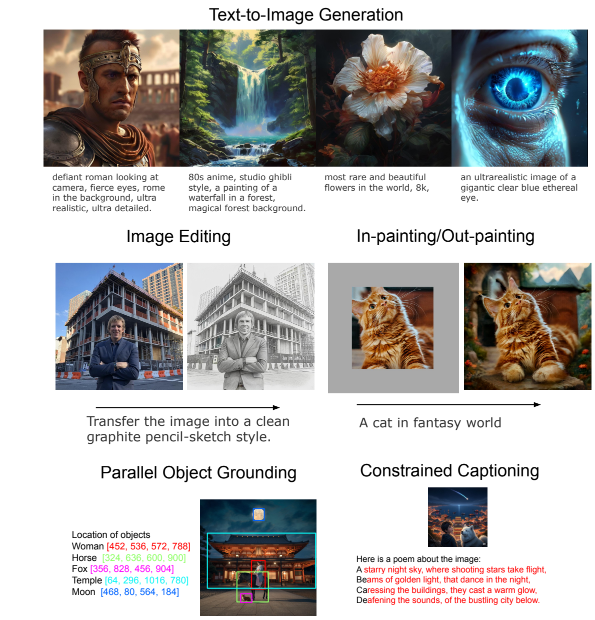


Figure 6. **Qualitative results.** Unlike semi-AR approaches like Block Diffusion, Sparse-LaViDa supports tasks requiring bidirectional context, such as inpainting/outpainting, parallel grounding, and constrained captioning. In text generation examples, colored regions denote masked tokens initialized for infilling.

Table 6. **Ablation Studies on Speed**. We report the speedup contribution of each key deigns of Sparse-LaViDa on T2I task.

Cache Prompt	Cache Res	Truncate Res	Latency ↓	Speedup ↑
			21.27	1.00
✓			16.43	1.29
	✓		18.87	1.13
		✓	17.93	1.19
✓	✓		14.09	1.51
	✓	✓	15.79	1.35
✓		✓	13.72	1.55
✓	✓	✓	10.86	1.96

caching prompt tokens, caching decoded response tokens, and truncating redundant tokens. We test all combinations of these components and report results in Table Tab. 6. As shown, enabling any single component improves efficiency, and combining all yields the maximum speedup.

Effect of register tokens. To study the impact of register tokens, we experiment with 0, 1, 32, and 64 registers and report results in Table Tab. 7. We evaluate GenEval and DPG scores as well as FID and HPS v3 metrics on MJHQ-30k. On GenEval, which evaluates high-level prompt alignment via object detection, removing register tokens causes little degradation. However, on DPG-bench, which evaluates fine-grained prompt alignment using a VQA model, the absence of register tokens leads to a larger performance drop. We also observe measurable differences in image quality

Table 7. **Ablation Studies on the Number of Registers.** We report text-to-image generation performance with different number of registers.

#Reg	GenEval ↑	DPG ↑	HPS v3 ↑	FID ↓
0	0.76	80.3	8.68	9.32
1	0.76	79.6	8.71	9.50
32	0.77	82.1	8.87	8.25
64	0.78	82.4	8.89	7.63

metrics (FID and HPS v3), suggesting that register tokens primarily enhance low-level visual detail rather than high-level structural coherence.

Table 8. **Ablation Studies on the Training Strategies.** We demonstrate the effectiveness of our training strategy through performance on GenEval and DPG-bench.

Model	GenEval	DPG
LaViDa-O [32]	0.77	81.8
Sparse-LaViDa	0.78	82.4
-No Step Causal Attention Mask	0.71	78.9
-No Training	0.24	47.9

Training strategy. We examine several design choices in our training pipeline, as summarized in Table Tab. 8. We find that applying the inference pipeline of Sparse-LaViDa to pretrained LaViDa-O without fine-tuning ("No Training") results in significant performance degradation on GenEval and DPG. Additionally, removing Step-Causal Attention Mask adversely affect the performance because of mismatched behaviors between training and inference.

5. Conclusion and Future works.

In conclusion, we propose Sparse-LaViDa, a novel parameterization for multi-modal MDMs. It offers significant speedup on a wide range of visual and text generation tasks such as text-to-image generation, image editing, and visual math reasoning without compromising generation quality. Despite promising results, Sparse-LaViDa has several limitations. First, while Sparse-LaViDa offers a significant speedup, it requires additional training. We emphasize that Sparse-LaViDa offers faster speedup than most aggressive KV caching strategy that caches all possible tokens (Tab. 6), which is an upper-bound for all heuristic-based training-free methods. Second, while in principle Sparse-LaViDa is just an efficient parameterization for the standard MDM and can be used to pre-train large models from scratch, we conducted our experiments in a post-trianing setup due to compute costs. In future we will explore additional scaling and train from scratch.

6. Acknowledgement

AG would like to acknowledge the support from Schmidt Sciences and NSF Career Award #2341040. SL is in part supported by Amazon Fellowship.

References

[1] 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. 2, 3

[2] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021. 2

[3] 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*, 2025. 5

[4] Victor Besnier, Mickael Chen, David Hurych, Eduardo Valle, and Matthieu Cord. Halton scheduler for masked generative image transformer. *arXiv preprint arXiv:2503.17076*, 2025. 4

[5] 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*, pages 18392–18402, 2023. 7

[6] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. https://github.com/kakaobrain/coyo-dataset, 2022. 6, 4

[7] 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*, pages 11315–11325, 2022. 2

[8] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 2

[9] 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*, 2025. 6, 4

[10] 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*, 2025. 6, 4

[11] Shuang Cheng, Yihan Bian, Dawei Liu, Yuhua Jiang, Yihao Liu, Linfeng Zhang, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, et al. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. *arXiv preprint arXiv:2510.06303*, 2025. 3

[12] 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. 1, 6, 7

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019. 2

[14] 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*, pages 12873–12883, 2021. 2

[15] 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. 6

[16] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. 6, 7

[17] 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. 6

[18] 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. 6, 4

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 2

[20] Minghui Hu, Chuanxia Zheng, Heliang Zheng, Tat-Jen Cham, Chaoyue Wang, Zuopeng Yang, Dacheng Tao, and Ponnuthurai N Suganthan. Unified discrete diffusion for simultaneous vision-language generation. *arXiv*, 2022. 4

[21] 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*, 5(7): 16, 2024. 6

[22] 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. 6, 4

[23] 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)*, pages 787–798, 2014. 4

[24] Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, et al. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025. 3

[25] 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*, pages 4015–4026, 2023. 6, 4

[26] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023. 7

[27] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 6

[28] 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. 7

[29] Dongxu Li, Junnan Li, and Steven Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. In *Advances in Neural Information Processing Systems*, pages 30146–30166. Curran Associates, Inc., 2023. 4

[30] Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2.5: Three insights towards enhancing aesthetic quality in text-to-image generation, 2024. 6

[31] Shufan Li, Harkanwar Singh, and Aditya Grover. Instructany2pix: Flexible visual editing via multimodal instruction following. *arXiv preprint arXiv:2312.06738*, 2023. 7

[32] 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:2509.19244*, 2025. 1, 2, 3, 4, 6, 7, 8

[33] 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*, 2025. 1, 3

[34] 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:2506.03147*, 2025. 7

[35] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024. 5

[36] 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. 7

[37] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer, 2024. 6

[38] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023. 2, 1

[39] 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. 6

[40] Chuofan Ma, Yi Jiang, Junfeng Wu, Jihan Yang, Xin Yu, Zehuan Yuan, BINGYUE PENG, and XIAOJUAN QI. Unitok: a unified tokenizer for visual generation and understanding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 1

[41] Xinyin Ma, Runpeng Yu, Gongfan Fang, and Xinchao Wang. dkv-cache: The cache for diffusion language models. *arXiv preprint arXiv:2505.15781*, 2025. 3

[42] Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15086–15095, 2025. 7

[43] Ahmed Masry, Do 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*, pages 2263–2279, Dublin, Ireland, 2022. Association for Computational Linguistics. 6, 7

[44] 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*, pages 2200–2209, 2021. 6, 7

[45] 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. 3, 1

[46] OpenAI. Dall·e 3. https://openai.com/index/dall-e-3/, 2023. 6

[47] OpenAI. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 7

[48] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 6

[49] 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*, pages 8748–8763. PmLR, 2021. 4

[50] 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. *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 4

[51] 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. 2, 3, 1

[52] Christoph Schuhmann. Laion-aesthetics. https://laion.ai/blog/laion-aesthetics/, 2022. Accessed: 2024 - 03 - 06. 4

[53] 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. 6, 4

[54] Qingyu Shi, Jinbin Bai, Zhuoran Zhao, Wenhao Chai, Kaidong Yu, Jianzong Wu, Shuangyong Song, Yunhai Tong, Xiangtai Li, Xuelong Li, et al. Muddit: Liberating generation beyond text-to-image with a unified discrete diffusion model. *arXiv preprint arXiv:2505.23606*, 2025. 1, 3

[55] Yuerong Song, Xiaoran Liu, Ruixiao Li, Zhigeng Liu, Zengfeng Huang, Qipeng Guo, Ziwei He, and Xipeng Qiu. Sparse-dllm: Accelerating diffusion llms with dynamic cache eviction. *arXiv preprint arXiv:2508.02558*, 2025. 3

[56] 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. 6, 4

[57] Xu Wang, Chenkai Xu, Yijie Jin, Jiachun Jin, Hao Zhang, and Zhijie Deng. Diffusion llms can do faster-than-ar inference via discrete diffusion forcing. *arXiv preprint arXiv:2508.09192*, 2025. 3, 6

[58] XuDong Wang, Shaolun Zhang, Shufan Li, Kehan Li, Konstantinos Kallidromitis, Yusuke Kato, Kazuki Kozuka, and Trevor Darrell. Segllm: Multi-round reasoning segmentation with large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. 5

[59] Yuhan Wang, Siwei Yang, Bingchen Zhao, Letian Zhang, Qing Liu, Yuyin Zhou, and Cihang Xie. Gpt-image-edit-1.5 m: A million-scale, gpt-generated image dataset. *arXiv preprint arXiv:2507.21033*, 2025. 6, 7, 4

[60] 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*, 2025. 3, 7

[61] 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:2506.18871*, 2025. 7

[62] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. 7

[63] Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, Song Han, and Yao Lu. VILA-u: a unified foundation model integrating visual understanding and generation. In *The Thirteenth International Conference on Learning Representations*, 2025. 1

[64] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13294–13304, 2025. 7

[65] 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. 6

[66] Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*, 2025. 1, 3, 6

[67] Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b, 2025. 3

[68] 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. 6, 7

[69] 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. 1

[70] Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26125–26135, 2025. 7

[71] Runpeng Yu, Xinyin Ma, and Xinchao Wang. Dimple: Discrete diffusion multimodal large language model with parallel decoding. *arXiv preprint arXiv:2505.16990*, 2025. 1, 3

[72] 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. 6, 7

[73] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023. 7

[74] 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. 6, 7

[75] Haozhe Zhao, Xiaojian Shawn Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. *Advances in Neural Information Processing Systems*, 37:3058–3093, 2024. 7

[76] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024. 1

[77] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. 5