CANVASFLASH: EFFICIENT AUTOREGRESSIVE IMAGE GENERATION WITH SEMANTIC REPRESENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long visual sequences impose substantial training and decoding costs on high-resolution autoregressive (AR) image generation. We propose CANVASFLASH, a novel approach that reduces this cost by exploiting the redundancy within visual representations. We depart from the reconstruction-oriented design in favor of a semantic image tokenizer that serves as a high-level information bottleneck, and delegate pixel rendering to a diffusion decoder. The resulting token distribution becomes highly concentrated: diverse tokens cluster in semantically informative regions, whereas broad regions share a common Canvas Token, indicating that only a fraction of the sequence is needed to characterize an image for semantic representation. Leveraging this sparsity, we propose Canvas-Sparse Training, which processes 41% of the tokens required by full-sequence training. We also propose Canvas-as-Draft Decoding, which uses the Canvas Token as a parameter-free draft and achieves up to $3.09\times$ speedup in AR decoding. With reinforcement learning, CANVASFLASH combines these efficiency gains with competitive image generation performance. More broadly, CANVASFLASH highlights the redundancy carried by reconstruction-oriented visual representations, and validates semantic tokenization as a practical route toward efficient autoregressive image generation.

1 INTRODUCTION

The success of Large Language Models (LLMs) has established autoregressive (AR) next-token prediction as a powerful paradigm (Brown et al., 2020; Achiam et al., 2023; Dubey et al., 2024), inspiring growing interest in autoregressive visual generation (Sun et al., 2024c; Wang et al., 2024a; Team, 2024; Geng et al., 2025). However, extending this paradigm to high-resolution images creates a fundamental bottleneck: with a patch size of 16×16 pixels, a 1024×1024 image produces $4{,}096$ visual tokens, imposing a substantial burden on both training and sequential decoding. This bottleneck motivates us to reconsider how information is organized in visual representations. Conventional image tokenizers (Van Den Oord et al., 2017; Esser et al., 2021) are optimized toward reconstruction fidelity. The visual sequence is therefore forced to perserve pixel-level variation with low information density, which raises a question: can autoregressive generation instead operate on a more compact representation with higher information utilization efficiency?

Motivated by this goal, we decompose image generation into two separate stages: semantic planning and pixel rendering. The autoregressive model first organizes the global semantics and spatial structure, and the diffusion decoder then synthesizes detailed visual appearance. This division of labor relieves the image tokenizer of the burden of pixel fidelity, allowing it to serve as a semantic bottleneck. Specifically, we supervise the image tokenizer with a multimodal understanding objective instead of pixel-level reconstruction, so that the discrete image tokens capture high-level features residing in a language-aligned semantic space.

Under this design, the token distribution of the image tokenizer exhibits a distinctive phenomenon. As shown in Fig. 1, although no explicit constraint is imposed, diverse token identities cluster in semantically informative regions, whereas broad regions consistently map to the same codebook entry across images. Visually, this organization resembles painting on a canvas: a bare canvas comes first, and pigment is then dabbed onto it to bring out the content. We therefore call the shared entry the **Canvas Token** and the resulting structure **Canvas Sparsity**. This pattern corroborates our design

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.

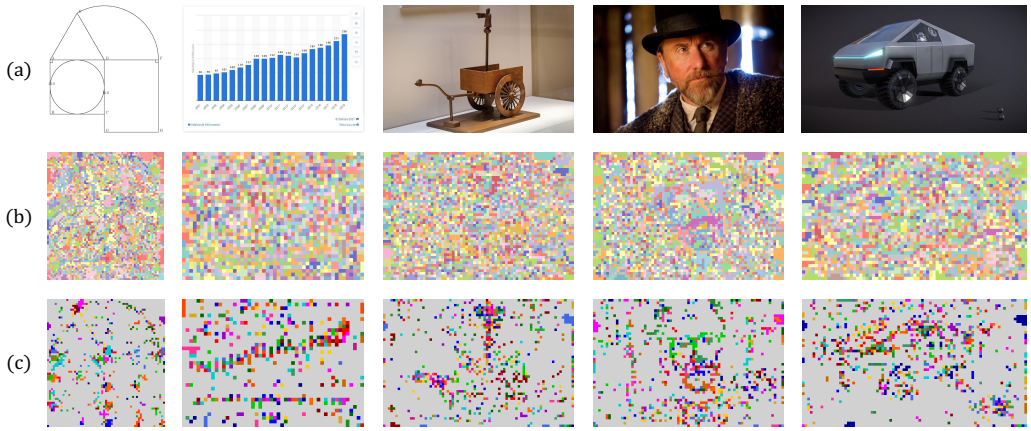

Figure 1: **Visualization of Canvas Sparsity.** The three rows show (a) the original images, (b) token maps from the X-Omni (Geng et al., 2025), with densely varying token identities, and (c) token maps from our tokenizer, where non-Canvas token diversity concentrates in semantically informative regions and broad regions map to a shared Canvas Token. Gray patches in (c) denote the same Canvas Token in all images. The shared code highlights repeated token structure; Fig. 4 illustrates the semantic content preserved when these tokens are rendered.

and indicates that semantic-level representation is intrinsically compact. A small fraction of positions suffices to characterize an image, while the remainder constitutes redundancy.

To exploit this pattern, we design tailored acceleration strategies for training and decoding. During training, we propose **Canvas-Sparse Training** to drop redundant Canvas Tokens without compromising generation quality, thereby reducing training cost. During inference, we design **Canvas-as-Draft Decoding** that treats Canvas Tokens as a parameter-free draft token to accelerate autoregressive decoding. Building on these efficiency gains, we further optimize generation quality through reinforcement learning. Our evaluations show that CANVASFLASH combines substantial training and decoding efficiency gains with competitive image generation quality.

Our contributions are summarized as follows:

- **Semantic Image tokenization.** We propose a decoupled framework and incorporate an image tokenizer with semantic supervision. Its emergent Canvas Sparsity indicates that only a fraction of the sequence is necessary to characterize an image for semantic representation.
- **Canvas-Sparse Training.** We propose a training acceleration strategy that drops redundant tokens from training sequences, substantially reducing training cost.
- **Canvas-as-Draft Decoding.** We propose a decoding acceleration strategy that treats Canvas Tokens as parameter-free drafts to accelerate autoregressive sampling without a separate draft model.
- **Competitive generation performance.** With reinforcement learning, CANVASFLASH combines competitive image generation performance with substantial training and decoding efficiency gains.

2 RELATED WORK

2.1 VISUAL TOKENIZATION AND AUTOREGRESSIVE GENERATION

Discrete AR image generation relies on visual tokenizers to map images into categorical sequences. Reconstruction-oriented tokenizers, represented by VQ-VAE (Van Den Oord et al., 2017; Razavi et al., 2019) and VQGAN (Esser et al., 2021), have been extended through residual quantization (Lee et al., 2022), finite-scalar quantization (Mentzer et al., 2024), lookup-free quantization (Yu et al., 2024), and ViT-based encoders (Yu et al., 2022a). Although effective for AR generation (Yu et al., 2022b; Sun et al., 2024a; Tian et al., 2024) and unified multimodal modeling (Sun et al., 2024c;b; Wang et al., 2024a; Dong et al., 2024; Wu et al., 2025d; Team, 2024), reconstruction objectives retain local appearance variations. Semantic tokenizers instead learn discrete representations under

high-level supervision, shifting emphasis from pixel fidelity to semantic content and structure (Wu et al., 2025c; Du et al., 2025; Team et al., 2026; Geng et al., 2025). We adopt the semantic supervision objective proposed in X-Omni (Geng et al., 2025), but initialize our tokenizer from the visual encoder of a pretrained VLM. Its text-aligned feature space provides a strong semantic prior that encourages the observed Canvas Sparsity. The resulting distribution concentrates diverse codes in semantically informative regions, while broad regions share the Canvas Token.

2.2 Visual Token Redundancy

A common premise in visual token reduction is that image sequences contain redundancy and that positions contribute unequally to semantic modeling. Masked Image Modeling (He et al., 2022; Xie et al., 2022) learns representations from heavily masked observations, while dynamic token pruning and merging (Rao et al., 2021; Bolya et al., 2023; Liang et al., 2022; Kong et al., 2022; Meng et al., 2022) reduce continuous visual features according to estimated relevance or similarity. Related VLM methods (Chen et al., 2024; Zhang et al., 2024; Shang et al., 2024; Ye et al., 2024; Xing et al., 2024; Wen et al., 2025) similarly reduce visual inputs with less contribution to language understanding. These approaches operate on the visual input, whereas we target the discrete output sequence of autoregressive generation. Moreover, Canvas Sparsity itself reveals where redundancy resides, dispensing with auxiliary modules for token selection.

2.3 Efficient Autoregressive Generation

Efficiency in autoregressive generation has been approached through architectural and decoding modifications. Sparse or windowed attention (Liu et al., 2021; Beltagy et al., 2020; Zaheer et al., 2020), linear attention (Katharopoulos et al., 2020), and state-space models (Gu & Dao, 2023; Zhu et al., 2024) reduce sequence-modeling cost, whereas masked prediction (Chang et al., 2022; 2023; Yu et al., 2023), next-scale prediction (Tian et al., 2024), and parallel AR decoding (Wang et al., 2024b) reorganize visual generation. Speculative decoding (Stern et al., 2018; Leviathan et al., 2023; Chen et al., 2023) reduces sequential model calls by verifying drafted tokens in parallel, with later work exploring prediction heads (Cai et al., 2024; Li et al., 2024a;b), tree-structured candidates (Miao et al., 2024), self-drafting (Zhang et al., 2023; Elhoushi et al., 2024), and image-specific proposals (Jang et al., 2024; Teng et al., 2024). In contrast, we exploit the prior provided by the image tokenizer and adopt the Canvas Token as a fixed draft, requiring no extra drafter. Moreover, Canvas Tokens tend to form extended contiguous spans, providing a structural regularity that favors the acceptance of an entire drafted window within a single verification pass.

3 Method

3.1 Overall Architecture

Following X-Omni (Geng et al., 2025), we separate image generation into two stages with distinct responsibilities, as illustrated in Fig. 2(b). The AR model predicts visual tokens that organize semantic content and spatial structure. Conditioned on these tokens, the diffusion decoder renders textures and pixel-level appearance. This division of labor allows the tokenizer to prioritize semantic content and spatial structure, while the diffusion decoder supplies pixel-level appearance.

Concretely, the semantic image tokenizer discretizes an image $\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$ into a grid of indices $\mathbf{z} \in \mathcal{V}_v^{h\times w}$ over a visual codebook $\mathcal{V}_v$, then raster-scans the grid into $(z_1, \ldots, z_N)$ with $N = hw$. Text and visual tokens are concatenated into a single autoregressive sequence while retaining their modality-specific vocabularies and embeddings. At inference, the AR model p_ϕ generates the visual sequence conditioned on the prompt. A diffusion-conditioning adapter projects the visual embeddings into spatial conditions for the diffusion decoder. The adapter and decoder are first fine-tuned for this conditioning interface and are then frozen during AR-model training.

3.2 Semantics-Oriented Image Tokenizer

As shown in Fig. 2(a), the image tokenizer consists of three components: a visual encoder f initialized from a pretrained VLM, a vector-quantization (VQ) module q, and a tokenizer-conditioning adapter

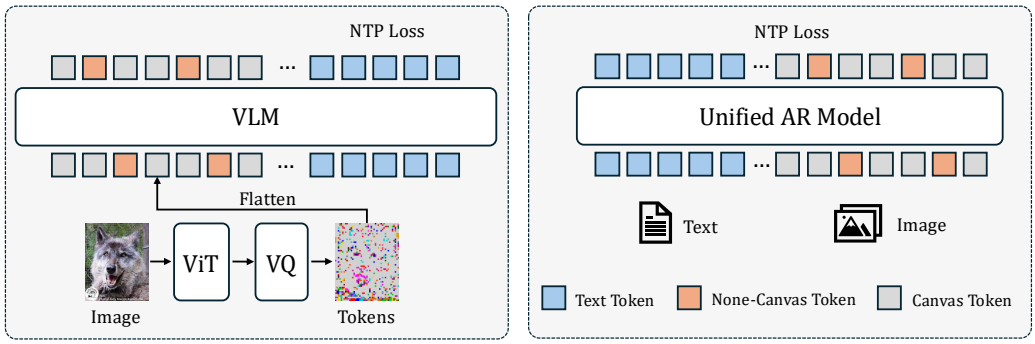

Figure 2: **Overall architecture of CANVASFLASH.** (a) The semantic image tokenizer starts from a frozen pretrained VLM visual encoder and supervised under a language-modeling objective. (b) Text and visual tokens form a unified sequence. The AR model predicts visual tokens encoding semantic content and spatial structure, and a diffusion decoder renders the visual tokens into pixels.

a. Given an image $\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$, the encoder produces visual features on an $h \times w$ grid, which we rasterize into $N = hw$ positions and denote by $(h_1, \ldots, h_N) = f(\mathbf{x})$.

The VQ module maps each feature to an entry of a codebook $E_v \in \mathbb{R}^{|\mathcal{V}_v|\times d_v}$. Since quantization is non-differentiable, we follow IBQ (**?**) and place the straight-through estimator on the index distribution rather than on the selected entry:

$$\begin{aligned} p_n &= \mathrm{softmax}\big(h_n E_v^\top\big)\,, \\ \tilde{e}_n &= \big(p_n + \mathrm{sg}[o_n - p_n]\big)\, E_v, \end{aligned} \tag{1}$$

where $z_n = \arg\max_k p_{n,k}$ is the selected index, o_n its one-hot vector, and $\mathrm{sg}[\cdot]$ the stop-gradient operator. The forward pass thus reduces to the quantized embedding e_{z_n}, while gradients flow back through p_n to the entire codebook, keeping it consistently utilized as the vocabulary scales.

The adapter then projects the quantized embeddings into the language-model input space, yielding the discrete indices $\mathbf{z} \in \mathcal{V}_v^{h\times w}$ together with the embedding sequence

$$\mathbf{v} = (v_1, \ldots, v_N) = (a(\tilde{e}_1), \ldots, a(\tilde{e}_N)) \in \mathbb{R}^{N\times d}, \tag{2}$$

where d is the language-model embedding dimension. The visual encoder remains frozen throughout tokenizer training, while the VQ module and adapter are trainable.

Let $\mathbf{y} = (y_1, \ldots, y_M)$ denote the textual annotation of $\mathbf{x}$, and let $t_m = E_t(y_m)$ be the embedding of its m-th token. We concatenate the visual embeddings and text embeddings into a single input sequence, with the visual embeddings serving as a prefix:

$$\mathbf{u} = (v_1, \ldots, v_N, t_1, \ldots, t_M). \tag{3}$$

The language model p_θ predicts each text token from the complete visual prefix and the preceding text tokens, giving the tokenizer-training objective

$$\mathcal{L}_{\mathrm{tok}} = -\sum_{m=1}^{M} \log p_\theta(y_m \mid \mathbf{v},\, y_{<m})\,. \tag{4}$$

Rather than supervising the tokenizer through pixel reconstruction, this objective anchors its discrete bottleneck to the language-aligned semantic space of the VLM. The resulting codes are optimized to retain image content that supports high-level understanding, leaving fine-grained appearance to the downstream rendering stage.

3.3 CANVAS-SPARSE TRAINING

For image generation, we prepend the text sequence $\mathbf{y} = (y_1, \ldots, y_M)$ to the visual sequence $\mathbf{z} = (z_1, \ldots, z_N)$, forming $\mathbf{s} = (y_1, \ldots, y_M, z_1, \ldots, z_N)$. The AR model p_ϕ processes this concatenated

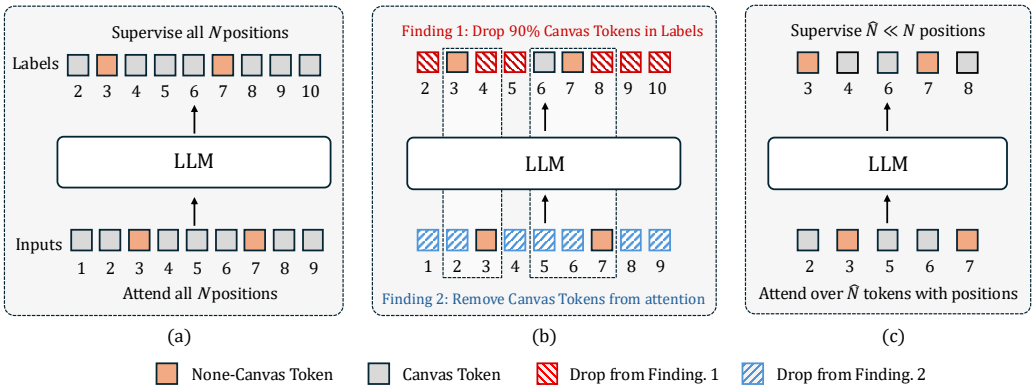

Figure 3: **Canvas-Sparse Training.** (a) Standard AR training supervises every target and includes every token as attention context. (b) Sparse Canvas Supervision marks $(1-\alpha)$ of Canvas-token targets as unsupervised (red hatches), while Canvas KV Masking excludes Canvas-token positions from attention key/value computation (blue hatches). (c) A position is removed when it serves as neither a supervised target nor key/value context; retained tokens keep their original position IDs.

sequence and applies the NTP loss only to its visual-token positions:

$$\mathcal{L}_{\mathrm{AR}} = -\sum_{j=M+1}^{M+N} \log p_\phi(s_j \mid s_{<j}). \tag{5}$$

During AR training, each visual token plays two roles: it serves as a prediction target supervised by its label, and its key/value states provide attention context for subsequent predictions. We examine the redundancy of Canvas Tokens in these two roles separately.

Finding 1: Supervising every Canvas-token is unnecessary. The recurrence of the shared label c across Canvas-token positions suggests substantial redundancy in their target supervision. To test this hypothesis, we retain an α fraction of Canvas-token positions in the label sequence and exclude the remaining positions from loss computation. We denote the indices marked as $\mathcal{U}_{\mathrm{drop}} \subseteq \{1,\dots,N\}$. As shown in Table 2, reducing Canvas-token supervision in the labels does not noticeably degrade generation quality, supporting our hypothesis.

Finding 2: Canvas Tokens provide limited attention context. The limited semantic variation across Canvas-token positions suggests that their key/value states provide largely redundant context for subsequent predictions. We further conduct a controlled experiment that excludes all Canvas-token positions from attention keys and values while retaining their query computations. Table 2 shows that this operation does not noticeably degrade generation quality. We therefore mark every Canvas-token position with $\varnothing_{\mathrm{KV}}$, indicating that its key/value states are omitted from attention computation, and denote the marked indices by $\mathcal{C}_{\mathrm{drop}} \subseteq \{1,\dots,N\}$.

Under sparse Canvas supervision and Canvas KV masking, positions carrying both markers serve neither of the two training roles. Using local visual indices, the removed and retained sets are

$$\mathcal{R} = \{1,\dots,N\} \setminus (\mathcal{U}_{\mathrm{drop}} \cap \mathcal{C}_{\mathrm{drop}}). \tag{6}$$

Canvas-Sparse Training removes the positions in $\mathcal{D}$, while the retained tokens keep their original position IDs to preserve the spatial geometry of the visual grid.

Prior correction. The selective labeling scheme reduces the effective frequency of Canvas-token targets during training, inducing a prior shift relative to the original label distribution. At inference, we compensate for this shift by adjusting the Canvas-token logit before sampling:

$$\ell'_c = \ell_c + \log(1/\alpha), \tag{7}$$

while leaving all non-Canvas logits unchanged. This parameter-free correction rescales the Canvas-to-non-Canvas odds by $1/\alpha$.

3.4 CANVAS-AS-DRAFT DECODING

Speculative decoding accelerates sequential generation by verifying multiple draft tokens in parallel. The pronounced concentration of the learned token distribution around the Canvas Token provides a parameter-free source of draft proposals. Given a prompt $\mathbf{y}$ and an accepted visual prefix $\mathbf{z}_{\leq t}$, we propose the Canvas code for the next K positions, i.e., $\hat{z}_{t+j} = c$ for $j = 1, \ldots, K$. A single evaluation of the prior-corrected target model then yields

$$q_j(\cdot) = p_\phi(\cdot \mid \mathbf{y}, \mathbf{z}_{\leq t}, \hat{z}_{t+1:t+j-1}), \qquad j = 1, \ldots, K+1. \tag{8}$$

Parallel verification draws one sample from each q_j and locates the first sample that differs from c. All preceding Canvas proposals are accepted, followed by the target-model sample at the first mismatched position. If no mismatch occurs within the draft window, all K proposals are accepted together with the additional sample from q_{K+1}. Every accepted proposal is verified against its realized prefix, preserving the sequential-sampling distribution of the same selectively supervised, prior-corrected target model under the same sampling configuration. Decoding still produces the full visual grid, but accepted Canvas runs allow one target-model evaluation to advance multiple positions.

3.5 REINFORCEMENT LEARNING WITH GRPO

The designs above reduce generation cost without changing the autoregressive framework. The model therefore remains amenable to reinforcement learning, which improves the token distribution rather than the decoding procedure and leaves the efficiency gains intact. We apply Group Relative Policy Optimizatio (GRPO) (Shao et al., 2024) to further improve the generation quality. For each prompt, the AR model samples a group of visual trajectories, which are rendered into images by the frozen diffusion decoder. The reward combines three complementary signals: an aesthetic reward for perceptual quality, an instruction-following reward for alignment with the input prompt, and a holistic reward for overall generation quality. In this way, reinforcement learning closes the loop between the AR model and the frozen diffusion decoder, and pulls the learned token distribution closer to that of real images.

4 EXPERIMENTS

4.1 IMPLEMENTATION

The image tokenizer is initialized from a pre-trained VLM derived from Hunyuan-7B (Team, 2025). We keep the visual encoder frozen during the training process of image tokenizer. The diffusion decoder is initialized from FLUX2-klein-9B (Black Forest Labs, 2026) and, together with a conditioning adapter, fine-tuned to render images conditioned on the semantic image tokens. After this alignment stage, the image tokenizer and diffusion decoder remain frozen throughout AR model optimization. The AR model is initialized from the language model of Hunyuan-VL-7B and trained under a unified next-token prediction objective with Canvas-Sparse Training applied as described in Sec. 3.3.

Aggregated over the full training corpus, Canvas-Sparse Training retains 41% of the full-sequence token count. We pre-train the AR model with 1.7T tokens, and an additional 17B tokens for SFT. The reward model is combined with HPSv3 (Ma et al., 2025) for visual aesthetics, Qwen3-VL-32B (Yang et al., 2025) for instruction following, and URM (Wang et al., 2025b) for holistic generation quality. Appendix E lists the hyperparameters.

4.2 QUALITATIVE ANALYSIS OF SEMANTIC RECONSTRUCTION

Figure 1 visualizes the image-token grids produced by our semantic tokenizer. To show how these tokens map back to images, we convert the discrete token indices into visual embeddings, project them through the diffusion-conditioning adapter, and feed the resulting spatial conditions to the frozen diffusion decoder. In the examples in Fig. 4, the rendered images preserve principal objects, semantic attributes, and spatial layouts while varying in fine-grained appearance. This behavior illustrates the intended division of labor: the tokens encode high-level content and structure, and the diffusion decoder renders pixel-level details.



Figure 4: **Semantic reconstruction of our image tokenizer.** (a) Original images. (b) Images rendered by projecting the image tokens into the frozen diffusion decoder. The reconstructions preserve the principal objects, semantic attributes, and spatial layouts, while fine-grained textures and appearance may differ because the tokenizer is not optimized for pixel-level reconstruction.

Table 1: General evaluation on image generation benchmarks. UGBench and UGBench-L denote the *English* and *English Long* splits of UniGenBench (Wang et al., 2025a), respectively.

Model	**Params**	**Image Generation**		
		DPG-Bench	**UGBench**	**UGBench-L**
Closed-source Models				
Nano Banana (DeepMind, 2025)	–	85.23	87.45	88.82
GPT-Image-1 (OpenAI, 2025)	–	85.20	92.77	92.63
Seedream 4.0 (Seedream et al., 2025)	–	88.25	87.30	89.77
Open-source Models				
Janus-Pro (Chen et al., 2025b)	7B	84.20	61.61	71.11
Show-o2 (Xie et al., 2025)	7B	86.14	62.73	70.33
BLIP3-o (Chen et al., 2025a)	7B + 1.4B	81.60	59.87	61.01
OmniGen2 (Wu et al., 2025b)	3B + 4B	83.57	63.09	71.39
UniWorld v1 (Lin et al., 2025)	7B + 12B	81.38	63.11	69.60
BAGEL (Deng et al., 2025)	14B	85.10	61.53	71.26
FLUX.1 [dev] (Labs, 2024)	12B	83.84	69.88	69.42
X-Omni (Geng et al., 2025)	7B + 12B	87.65	53.77	67.00
Lumina-DiMOO (Xin et al., 2025)	8B	86.04	71.12	71.81
Z-Image-Turbo (Tongyi-MAI, 2025)	4B + 6B	85.15	71.40	80.72
FLUX.2 [dev] (Labs, 2025)	32B	88.03	84.76	90.31
Qwen-Image (Wu et al., 2025a)	7B + 20B	88.32	78.81	83.94
GLM-Image (Z.ai, 2026)	9B + 7B	84.78	67.23	75.48
CANVASFLASH	7B + 9B	89.69	86.31	88.66

4.3 EVALUATION

We evaluate image generation on DPG-Bench (Hu et al., 2024) and the *English* and *English Long* splits of UniGenBench (Wang et al., 2025a). Unless stated otherwise, baseline results are taken from the original publications or official leaderboards. We use top-$p = 0.95$ sampling with a classifier-free guidance scale of 2.5 and generate images at 1024×1024 resolution.

Table 1 shows that the complete system delivers strong generation quality. CANVASFLASH obtains 89.69 on DPG-Bench, the highest score among all models in the table, exceeding the strongest listed closed-source and open-source results by 1.44 and 1.37 points, respectively. On the *English* split of UniGenBench, our score of 86.31 is the best among open-source models and improves over FLUX.2 [dev] by 1.55 points. On the more demanding *English Long* split, CANVASFLASH

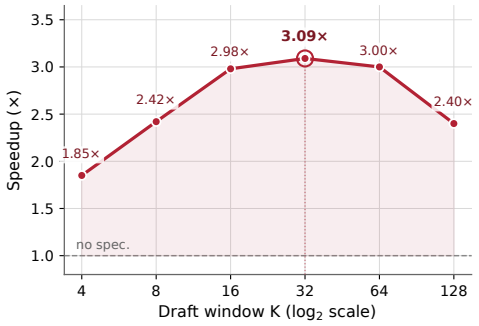

Figure 5: **AR token-decoding speedup vs. draft window** K at 1024^2, excluding diffusion rendering. Peak $3.09\times$ at $K{=}32$.

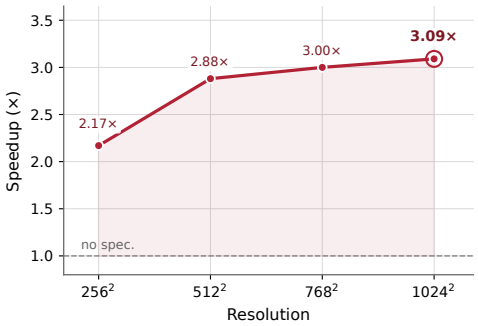

Figure 6: **AR token-decoding speedup vs. resolution** at $K{=}32$, excluding diffusion rendering. Speedup increases with resolution.

reaches 88.66, remaining competitive with leading systems. These results position Canvas Sparsity as a computational mechanism rather than a source of quality gains: it preserves the capabilities established by the complete system while substantially reducing training and decoding cost.

Table 2: Ablation of Canvas-token removal during AR training. **Label** marks Canvas-token positions that are excluded from loss computation, while **Attention KV** marks those excluded from attention keys and values. Only positions marked by both operations are removed from the sequence.

Label	**Attention KV**	DPG-Bench	Retained Tokens
		77.46	1.00×
✓		77.41	1.00×
	✓	78.17	1.00×
✓	✓	79.29	0.45×

4.4 Ablation Study on Canvas Sparsity Training

Table 2 evaluates three variants against a full-sequence baseline to isolate the two findings from Sec. 3.3. Retaining supervision on 10% of Canvas-token labels yields 77.41 on DPG-Bench, compared with 77.46 under full label supervision, indicating that reducing Canvas-token supervision maintains generation quality. Separately, excluding all Canvas-token positions from attention keys and values yields 78.17, again comparable to the baseline, indicating that Canvas Tokens contribute little context to subsequent predictions.

Combining the two operations, Canvas-Sparse Training retains 45% of the tokens on the small-scale dataset and reaches 79.29 under the same training-token budget. The slight improvement is attributable to the shortened sequences. Under the same token budget, they admit more training samples and thereby expose the model to broader supervision. Taken together, these results indicate that dropping redundant Canvas Tokens during training does not compromise generation quality. Canvas-Sparse Training is therefore a sound design.

4.5 Ablation Study on Decoding Parameters

Canvas-as-Draft Decoding accelerates only the AR token-decoding stage, so all reported speedups exclude diffusion rendering. We define acceptance rate as the fraction of proposed Canvas drafts accepted by the target model and average accepted length as the mean number of accepted Canvas drafts per verification step.

Draft-window size. Table 3 and Fig. 5 vary the draft window K, which balances parallel verification against acceptance. As K increases from 4 to 128, the average accepted length grows from 2.57 to 5.83, while the acceptance rate falls from 64.15% to 4.55%. Accepted length begins to saturate beyond $K{=}32$, so larger windows add verification work without advancing proportionally more

Table 3: Window-size ablation for Canvas-as-Draft Decoding at 1024×1024. Speedup measures the AR token-decoding stage relative to standard decoding ($K{=}0$)

Window Size K	**Speedup**	**Acceptance Rate**	**Avg. Accepted Length**
0	$1.00\times$	–	–
8	$2.42\times$	51.19%	4.10
16	$2.98\times$	32.36%	5.18
32	$\mathbf{3.09}\times$	17.66%	5.65
64	$3.00\times$	8.93%	5.72
128	$2.40\times$	4.55%	5.83

positions. The resulting speedup peaks at $3.09\times$ for $K{=}32$ and decreases to $2.40\times$ at $K{=}128$, making $K{=}32$ the best-performing window among the tested settings.

Image resolution. Figure 6 varies image resolution while fixing $K{=}32$. Speedup increases from $2.17\times$ at 256×256 to $3.09\times$ at 1024×1024. Thus, with $K{=}32$, Canvas-as-Draft Decoding provides larger measured speedups at higher tested resolutions.

Table 4: Effect of reinforcement learning on DPG-Bench. Results include the standalone diffusion decoder and CANVASFLASH before and after GRPO.

Model	DPG-Bench
FLUX2-klein-9B (Black Forest Labs, 2026)	86.55
CANVASFLASH (SFT)	83.95
CANVASFLASH (RL)	89.69

4.6 EFFECT OF REINFORCEMENT LEARNING

Table 4 shows the effect of applying GRPO within CANVASFLASH. DPG-Bench increases from 83.95 after SFT to 89.69, exceeding the performance of the standalone FLUX2-klein-9B model. The complete framework thus delivers stronger performance than the decoder alone. This comparison highlights the value of integrating semantic autoregressive modeling and reinforcement learning with diffusion rendering, beyond simply relying on the pretrained decoder.

5 CONCLUSION

To reduce the training and decoding cost of high-resolution AR image generation, we propose CANVASFLASH, which decomposes image generation into semantic planning and pixel rendering. Relieved of pixel fidelity, the image tokenizer serves as a semantic bottleneck and is accordingly supervised by multimodal understanding rather than pixel reconstruction. Its token distribution turns out to be highly concentrated in semantically informative regions, leaving broad positions in the sequence redundant. Canvas-Sparse Training and Canvas-as-Draft Decoding exploit this property to accelerate training and decoding, and the complete system achieves competitive generation quality.

More broadly, this work points to a promising discrete image representation, whose token distribution exhibits a distinctive sparsity pattern that can be translated into efficiency gains. Two directions naturally follow. First, the token distribution of the image tokenizer is currently shaped without explicit constraint, leaving open how such sparsity emerges and how the semantic granularity of the discrete representation can be controlled in a finer-grained manner. Second, the sparsity of visual signals is not specific to image generation. The same principle may extend to a broader range of image and video applications, enabling efficient understanding and generation over substantially long visual sequences.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Black Forest Labs. FLUX.2 [klein]: Towards interactive visual intelligence. `https://bfl.ai/blog/flux2-klein-towards-interactive-visual-intelligence`, 2026.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *ICLR*, 2023.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. Medusa: Simple llm inference acceleration framework with multiple decoding heads. In *ICML*, 2024.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *CVPR*, pp. 11315–11325, 2022.

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.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

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.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *ECCV*, pp. 19–35. Springer, 2024.

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*, 2025b.

Google DeepMind. Nano banana: State-of-the-art image generation and editing models, built on gemini. `https://deepmind.google/models/gemini-image/`, 2025.

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

Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, et al. Dreamllm: Synergistic multimodal comprehension and creation. In *ICLR*, 2024.

Sinan Du, Jiahao Guo, Bo Li, Shuhao Cui, Zhengzhuo Xu, Yifu Luo, Yongxian Wei, Kun Gai, Xinggang Wang, Kai Wu, et al. Vqrae: Representation quantization autoencoders for multimodal understanding, generation and reconstruction. *arXiv preprint arXiv:2511.23386*, 2025.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, et al. Layerskip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12622–12642, 2024.

Patrick Esser, Robin Rombach, and Björn 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.

Zigang Geng, Yibing Wang, Yeyao Ma, Chen Li, Yongming Rao, Shuyang Gu, Zhao Zhong, Qinglin Lu, Han Hu, Xiaosong Zhang, et al. X-omni: Reinforcement learning makes discrete autoregressive image generative models great again. *arXiv preprint arXiv:2507.22058*, 2025.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pp. 16000–16009, 2022.

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.

Doohyuk Jang, Sihwan Park, June Yong Yang, Yeonsung Jung, Jihun Yun, Souvik Kundu, Sung-Yub Kim, and Eunho Yang. Lantern: Accelerating visual autoregressive models with relaxed speculative decoding. *arXiv preprint arXiv:2410.03355*, 2024.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, pp. 5156–5165, 2020.

Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *ECCV*, pp. 620–640. Springer, 2022.

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

Black Forest Labs. Flux.2 [dev]. `https://bfl.ai/blog/flux-2`, 2025.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *CVPR*, pp. 11523–11532, 2022.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pp. 19274–19286, 2023.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle: Speculative sampling requires rethinking feature uncertainty. In *ICML*, 2024a.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle-2: Faster inference of language models with dynamic draft trees. *arXiv preprint arXiv:2406.16858*, 2024b.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *ICLR*, 2022.

Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, et al. Uniworld-v1: High-resolution semantic encoders for unified visual understanding and generation. *arXiv preprint arXiv:2506.03147*, 2025.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pp. 10012–10022, 2021.

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.

Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. Adavit: Adaptive vision transformers for efficient image recognition. In *CVPR*, pp. 12309–12318, 2022.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. In *ICLR*, 2024.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, et al. Specinfer: Accelerating large language model serving with tree-based speculative inference and verification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pp. 932–949, 2024.

OpenAI. Addendum to gpt-4o system card: 4o image generation. `https://openai.com/index/gpt-4o-image-generation-system-card-addendum/`, 2025.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *NeurIPS*, pp. 13937–13949, 2021.

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *NeurIPS*, 2019.

Team Seedream, Yunpeng Chen, Yu Gao, Lixue Gong, Meng Guo, Qiushan Guo, Zhiyao Guo, Xiaoxia Hou, Weilin Huang, Yixuan Huang, et al. Seedream 4.0: Toward next-generation multimodal image generation. *arXiv preprint arXiv:2509.20427*, 2025.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. Blockwise parallel decoding for deep autoregressive models. *NeurIPS*, 2018.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024a.

Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. In *CVPR*, pp. 14398–14409, 2024b.

Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Emu: Generative pretraining in multimodality. In *ICLR*, 2024c.

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

Meituan LongCat Team, Bin Xiao, Chao Wang, Chengjiang Li, Chi Zhang, Chong Peng, Hang Yu, Hao Yang, Haonan Yan, Haoze Sun, et al. Longcat-next: Lexicalizing modalities as discrete tokens. *arXiv preprint arXiv:2603.27538*, 2026.

Tencent Hunyuan Team. Tencent-hunyuan-7b. https://github.com/Tencent-Hunyuan/Hunyuan-7B, 2025.

Yao Teng, Han Shi, Xian Liu, Xuefei Ning, Guohao Dai, Yu Wang, Zhenguo Li, and Xihui Liu. Accelerating auto-regressive text-to-image generation with training-free speculative jacobi decoding. *arXiv preprint arXiv:2410.01699*, 2024.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *NeurIPS*, pp. 84839–84865, 2024.

Tongyi-MAI. Z-Image. https://github.com/Tongyi-MAI/Z-Image, 2025.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 2017.

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*, 2024a.

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

Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025b.

Yuqing Wang, Shuhuai Ren, Zhijie Lin, Yujin Han, Haoyuan Guo, Zhenheng Yang, Difan Zou, Jiashi Feng, and Xihui Liu. Parallelized autoregressive visual generation. *arXiv preprint arXiv:2412.15119*, 2024b.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for "important tokens" in multimodal language models: Duplication matters more. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9972–9991, 2025.

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.

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ze Liu, Ziyi Xia, Chaofan Li, Haoge Deng, Jiahao Wang, Kun Luo, Bo Zhang, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025b.

Shengqiong Wu, Hao Fei, Xiangtai Li, Jiayi Ji, Hanwang Zhang, Tat-Seng Chua, and Shuicheng Yan. Towards semantic equivalence of tokenization in multimodal llm. In *International Conference on Learning Representations*, volume 2025, pp. 93334–93358, 2025c.

Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. In *ICLR*, 2025d.

Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. In *NeurIPS*, 2025.

Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *CVPR*, pp. 9653–9663, 2022.

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

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, Ying Shan, and Yansong Tang. Voco-llama: Towards vision compression with large language models. *arXiv preprint arXiv:2406.12275*, 2024.

Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. In *ICLR*, 2022a.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022b.

Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10459–10469, 2023.

Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion – tokenizer is key to visual generation. In *ICLR*, 2024.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *NeurIPS*, pp. 17283–17297, 2020.

Z.ai. Glm-image: Auto-regressive for dense-knowledge and high-fidelity image generation. `https://z.ai/blog/glm-image`, 2026.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. Draft & verify: Lossless large language model acceleration via self-speculative decoding. *arXiv preprint arXiv:2309.08168*, 2023.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *ICML*, 2024.

A AI ASSISTANCE

We used generative AI tools only to retrieve and identify related work and to polish author-written text for clarity and readability. We did not use generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proofs, assist in writing proofs, or propose or refine hypotheses. We also did not use them to design or provide feedback on research methodology or experiments, implement methods, translate text, clean or reformat datasets, conduct qualitative or thematic data analysis, or interpret results.We reviewed all AI-assisted work. Specifically, we manually verified the authenticity and relevance of the references used and checked the revised text for accuracy and preservation of the intended meaning. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

B TOKEN MAPS FROM OTHER TOKENIZER

We repeat the tokenizer training procedure on Qwen2.5-VL-Instruct-7B (Bai et al., 2025). As shown in Fig. 7, a Canvas-like distribution also emerges after small-scale training, with broad regions sharing a dominant code. Its recurrence across backbones suggests that Canvas Sparsity reflects a shared semantic structure inherited from VLM features closely aligned with the language space. This connection supports the generality of our tokenization approach beyond a particular backbone.

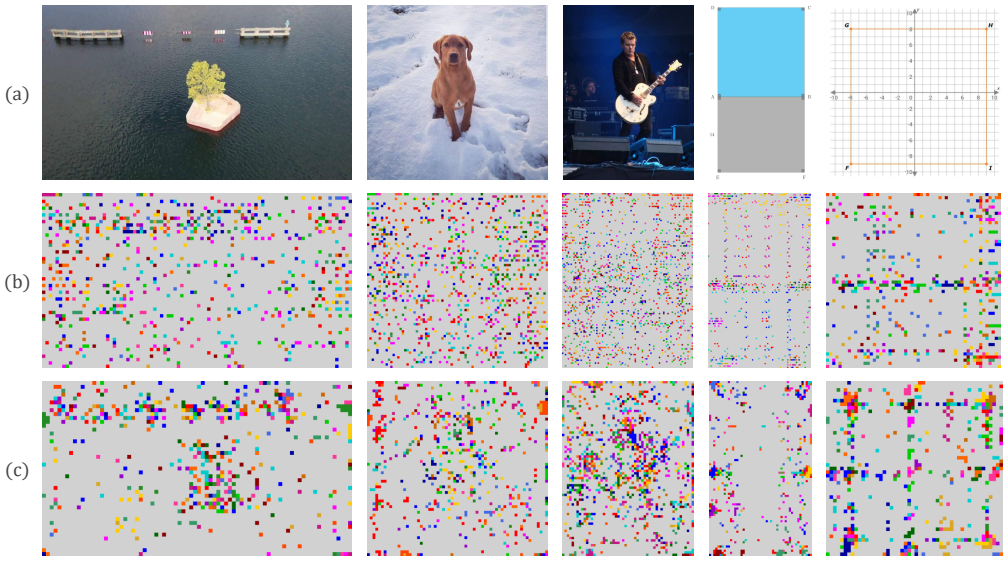

Figure 7: Visualization of image tokens. (a) Original images. (b) Token maps from the image tokenizer trained from Qwen2.5-VL. (c) Token maps from our image tokenizer.

C IMAGE EDITING

Beyond text-to-image generation, CANVASFLASH also supports image editing. Given a source image and a textual instruction, the model generates an edited image reflecting the requested change. Figure 8 presents qualitative examples of this extended capability.

D INTERLEAVED GENERATION

Figure 9 shows an interleaved trajectory in which CANVASFLASH generates text and intermediate images, with each step conditioned on the preceding multimodal history. This example highlights the potential of CANVASFLASH for long-context multimodal generation, where text and images accumulate within a shared autoregressive sequence. Extending Canvas-based acceleration to longer interleaved trajectories offers a promising direction for efficient multi-step visual generation.



Figure 8: Qualitative cases of image editing.

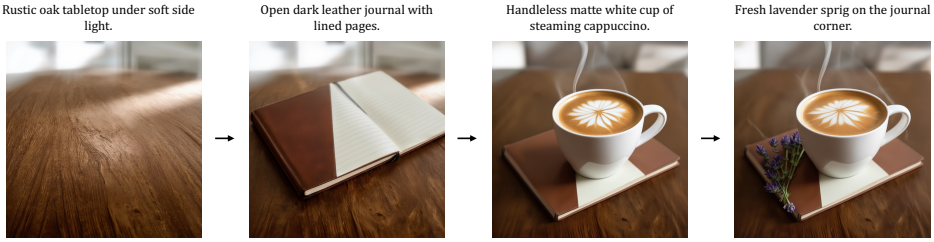

Figure 9: **Example of interleaved generation.** Given one prompt, CANVASFLASH generates interleaved text and intermediate images in a single autoregressive trajectory. The displayed text is abbreviated for visualization.

E TRAINING DETAILS

We provide additional training hyperparameters for the image tokenizer and the unified autoregressive model in Tab. 5 and Tab. 6, respectively. The tables report the sequence and context lengths, training steps, global batch sizes, trainable modules, optimizer settings, and learning rate schedules.

Table 5: Hyperparameters of the image tokenizer training procedure. From Stage 2, we apply different learning rate schedules for the image tokenizer (VQ) module and the LLM module.

Hyperparameter	**Stage 1**	**Stage 2**	**Stage 3**
Sequence Length	3600	3600	3600
Training Steps	10K	36K	6K
ViT	Frozen	Frozen	Frozen
VQ	Trainable	Trainable	Trainable
LLM	Frozen	Trainable	Trainable
Optimizer	Adam	Adam	Adam
Warmup Ratio	0.03	0.03	0.03
Weight Decay	0.0	0.0	0.0
Gradient Clipping	1.0	1.0	1.0
VQ Learning Rate Schedule	5×10^{-4}	1×10^{-4}	$1 \times 10^{-4} \rightarrow 0$
LLM Learning Rate Schedule	-	1×10^{-6}	$1 \times 10^{-6} \rightarrow 0$

Table 6: Hyperparameters of the two-stage training procedure for unified AR modeling. Stage 1 consists of a constant-learning-rate phase followed by a cosine learning-rate decay phase.

Hyperparameter	**Stage 1 (Pretrain)**	**Stage 2 (SFT)**
Context Length	8192	16384
Training Steps	100K + 3.5K	1K
Global Batch Size	2048	1024
Tensor Parallel Size	2	2
Context Parallel Size	1	2
Optimizer	Adam	Adam
Warmup Steps	500	100
Weight Decay	0.1	0.1
Gradient Clipping	1.0	1.0
Learning Rate Schedule	2×10^{-4} for first 100K steps $2 \times 10^{-4} \rightarrow 4 \times 10^{-5}$ for next 3.5K steps	$4 \times 10^{-5} \rightarrow 0$

F Qualitative Cases

Figure 10 illustrates more qualitative results generated by our model.



Figure 10: Qualitative cases of image generation.