FLASHAR: EFFICIENT POST-TRAINING ACCELERATION FOR AUTOREGRESSIVE IMAGE GENERATION

Anonymous authors
Paper under double-blind review

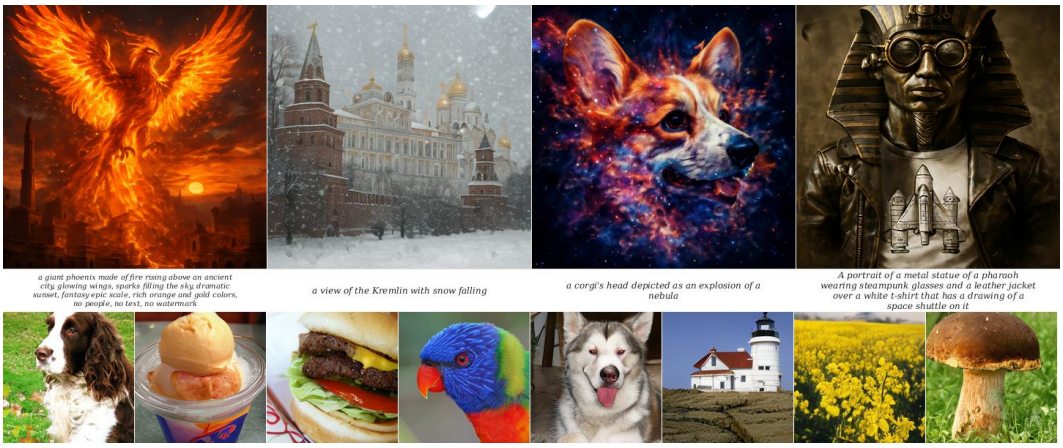


Figure 1: **Generated samples from FlashAR.** The first row shows 512×512 text-guided generation results, while the second row presents class-conditional generation samples at 384×384 and 256×256 resolutions.

ABSTRACT

Large-scale autoregressive models have demonstrated remarkable capabilities in image generation. However, their sequential raster-scan decoding relies on strictly next-token prediction, making inference prohibitively expensive. Existing acceleration methods typically either introduce entirely new generation paradigms that necessitate costly pre-training from scratch, or enable parallel generation at the expense of a training-inference gap or altered prediction objectives. In this paper, we introduce **FlashAR**, a lightweight post-training adaptation framework that efficiently adapts a pre-trained raster-scan autoregressive model into a highly parallel generator based on two-way next-token prediction. Our key insight is that effective adaptation should minimize modifications to the pre-trained model's original training objective to preserve its learned prior. Accordingly, we retain the original AR head as a horizontal head for row-wise prediction and introduce a complementary, lightweight vertical head for column-wise prediction. To facilitate efficient adaptation, we branch the vertical head from a pre-final layer rather than the final layer, bypassing the inherent horizontal head bias. Moreover, since horizontal and vertical predictions capture complementary dependencies whose relative importance varies across target positions, we employ a learnable fusion gate to dynamically combine the two predictions at each position. To further reduce adaptation cost, we propose a two-stage adaptation pipeline: the vertical head is first initialized through adaptation from the pre-trained autoregressive model before jointly fine-tuned with backbone to adapt to the new decoding paradigm. Extensive experiments on LlamaGen and Emu3.5-Image show that FlashAR achieves a **13.03**$\times$ speedup for 512×512 generation on Emu3.5-Image using approximately **0.05%** of its pre-training tokens.

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.

1 INTRODUCTION

Autoregressive (AR) models (Cui et al., 2025; Wang et al., 2024a; Zhipu AI, 2026; Liu et al., 2024; Xin et al., 2025; NextStep Team et al., 2025; Wang et al., 2025a; Geng et al., 2025) have emerged as a powerful paradigm for high-fidelity image generation. By representing visual data as discrete token sequences (Van Den Oord et al., 2017; Esser et al., 2021), these models achieve strong scalability and generation quality. However, these models remain fundamentally constrained by raster-scan decoding, which generates image tokens sequentially from left to right and top to bottom. As a result, the decoding latency grows linearly with the number of image tokens, making high-resolution generation prohibitively slow. Moreover, predicting only one token at each step prevents AR decoding from effectively utilizing the parallel computing capabilities of modern GPUs.

Existing acceleration methods have attempted to mitigate this bottleneck, but they often introduce new limitations. One line of work redesigns the generation paradigm, for example by changing the token prediction order (He et al., 2025; Wang et al., 2025b; Zhang et al., 2025; Li et al., 2025; Yu et al., 2025) or adopting multi-scale autoregressive generation (Tian et al., 2024; Han et al., 2025; Tang et al., 2024). Although such methods can substantially reduce decoding latency, they usually require costly pre-training from scratch and cannot directly reuse the large repository of existing pre-trained raster-scan AR models. Another line of work enables parallel generation through post-training adaptation, such as discrete diffusion adaptation (Cui et al., 2025). However, these approaches modify the original prediction objective and introduce a discrepancy between pre-training and inference. This dilemma raises a critical open question: *Can we transform a pre-trained raster-AR model into a highly parallel generator while inheriting its powerful generative capabilities and keeping the post-training overhead minimal?*

To address these challenges, we propose **FlashAR**, a lightweight post-training adaptation framework that efficiently adapts a pre-trained raster-scan autoregressive model into a highly parallel generator by reusing its horizontal predictor and adding vertical and fused prediction objectives. As shown in Figure 2, we retain the original AR head as a horizontal head for row-wise prediction and introduce a lightweight vertical head for column-wise prediction, which enables the model to support parallel decoding under a two-way next-token prediction structure.

A key challenge in introducing vertical prediction is that the pre-trained model is inherently biased toward the original horizontal raster-scan objective and directly attaching a vertical head to the final layer is therefore suboptimal. To facilitate efficient adaptation and bypass this inherent horizontal head bias, we branch the vertical head from a pre-final layer rather than from the final layer, allowing the vertical pathway to access richer representations.

Moreover, horizontal and vertical predictions capture complementary directional dependencies, and their relative importance varies across spatial positions. A fixed combination of the two predictions is therefore insufficient and may introduce spatial inconsistency or context conflicts. To address this issue, FlashAR employs a learnable fusion gate that dynamically combines horizontal and vertical predictions at each target position, allowing the model to exploit their complementarity. To reduce adaptation cost, we first initialize the vertical head through adaptation of the pre-trained model, then jointly fine-tune it with the backbone for two-way decoding. Our inference pipeline combines FlexAttention-compiled sparse 2D proximity masks with batched KV-cache updates to reduce memory and kernel launch overheads. Extensive experiments on LlamaGen and Emu3.5 demonstrate that FlashAR achieves a 13.03$\times$ speedup for 512×512 generation on Emu3.5-Image using approximately 0.05% of its pre-training tokens.

In summary, our main contributions are as follows:

- We propose **FlashAR**, a lightweight post-training acceleration framework that efficiently transforms pre-trained raster-scan AR models into highly parallel generators, largely retaining their generative capabilities without the prohibitive cost of training from scratch.
- We introduce a dual-head pre-final branching architecture that resolves the tension between preserving a pre-trained model's horizontal prior and introducing orthogonal vertical prediction. A learnable fusion gate further allows the model to dynamically arbitrate between horizontal and vertical signals depending on spatial context.
- We establish a two-stage post-training pipeline that ensures rapid adaptation and fast convergence.

- We demonstrate the effectiveness of FlashAR across both class-conditional and text-to-image models, achieving a 13.03× speedup on Emu3.5-Image with limited post-training and competitive generation quality.

2 Related Work

Autoregressive visual generation. Building on the remarkable success of large language models (LLMs) (Touvron et al., 2023; Bai et al., 2023; Yang et al., 2025; Gemini Team et al., 2023), autoregressive (AR) architectures have rapidly emerged as a compelling paradigm for visual and multimodal generation (Duan et al., 2026). These approaches typically rely on learned visual tokenizers to quantize continuous images into discrete latent tokens. By flattening these 2D token maps into 1D sequences, visual generation is naturally cast as a next-token prediction task under a strict causal objective. Early pioneering works, including PixelCNN (Van den Oord et al., 2016), iGPT (Chen et al., 2020) and Parti (Yu et al., 2022), demonstrated the strong potential of this formulation. More recently, driven by well-established scaling laws, large-scale models—such as Emu (Wang et al., 2024a; Cui et al., 2025), NextStep-1 (NextStep Team et al., 2025), Lumina-mGPT (Liu et al., 2024; Xin et al., 2025) and GLM-Image (Zhipu AI, 2026)—have substantially advanced the paradigm. By unifying visual modeling under a rigorous causal objective, these models effectively capture complex long-range spatial dependencies, achieving quality that rivals or surpasses contemporary diffusion-based approaches. Despite these advances, sequential raster-scan decoding remains a latency bottleneck, particularly for high-resolution image generation.

Efficient and parallel autoregressive decoding. To mitigate the inference latency of standard visual AR models, recent work has explored several parallelization strategies. One line of research accelerates decoding by departing from strict raster-scan token ordering. For instance, VAR (Tian et al., 2024; Han et al., 2025; Tang et al., 2024) reformulates generation as coarse-to-fine "next-scale prediction", while NAR (He et al., 2025) exploits spatial locality through "next-neighbor prediction". PAR (Wang et al., 2025b) partitions image tokens into subsets and applies the standard next-token prediction paradigm within each subset. Although these structural changes successfully reduce sequential dependencies, they require bidirectional attention mechanisms and thus must be trained from scratch—making them fundamentally incompatible with existing pre-trained raster-AR models. A second line bridges AR and diffusion frameworks via discrete diffusion adaptation (Deng et al., 2025; Gat et al., 2024; Arriola et al., 2025; Shi et al., 2025). Methods such as Emu3.5 (Cui et al., 2025) replace lengthy AR chains with parallel refinement over noisy token blocks. While this improves throughput, imposing a diffusion objective fundamentally alters the causal structure of the AR backbone, necessitating multi-stage fine-tuning and often compromising the fine-grained structural consistency that strict causal modeling naturally provides. A third alternative involves speculative decoding (Teng et al., 2024; Jang et al., 2024; Wang et al., 2024b), which serves as a training-free acceleration plug-in. However, the practical speedup of speculative decoding is fundamentally constrained by the acceptance rate of the draft model, typically yielding marginal gains compared to architectural parallelization. Therefore, we do not include speculative decoding as a primary baseline for comparison. In contrast, FlashAR focuses on accelerating existing raster-scan AR models through lightweight post-training. Building on the two-direction decoding of NAR (He et al., 2025), it introduces a pre-final vertical branch and a learnable fusion gate to enable parallel generation while retaining the pre-trained horizontal predictor.

3 Preliminaries

3.1 Standard Raster-Scan Autoregressive Image Generation

Given a generation condition c and a discrete image token grid $Y = \{y_{p,q}\}_{p=0,q=0}^{H-1,W-1}$ of size $H \times W$, standard autoregressive image generation (Sun et al., 2024; Wang et al., 2024a) flattens the 2D token grid into a 1D sequence using raster-scan order. The conditional distribution is factorized as

$$p(Y \mid c) = \prod_{i=0}^{HW-1} p(y_i \mid y_{<i}, c). \tag{1}$$

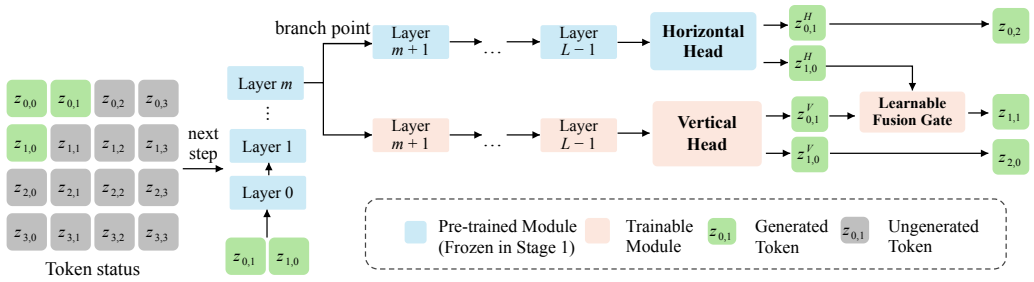


Figure 2: Overview of the FlashAR framework. Initialized from a pre-trained raster-scan autoregressive model, the architecture incorporates a pre-final branching module for the vertical head and a learnable fusion gate, facilitating efficient adaptation to parallel generation.

While this formulation is expressive, it enforces strictly sequential decoding over all HW image tokens. As a result, the inference cost grows proportionally with the total number of visual tokens, which becomes increasingly expensive for high-resolution image synthesis.

3.2 Diagonal-Step Factorization

Following NAR (He et al., 2025), we factorize the autoregressive image generation into two orthogonal directions, where the original decoding head serves as next-token predictor for the row-wise prediction and an additional vertical head is introduced for column-wise prediction. Consequently, it partitions the 2D grids into

$$\mathcal{D}_t = \{y_{p,q} \mid p + q = t\}, \qquad t = 0, 1, \ldots, H + W - 2. \tag{2}$$

This yields a diagonal-step factorization of the image distribution:

$$p(Y \mid c) = \prod_{t=0}^{H+W-2} p(\mathcal{D}_t \mid \mathcal{D}_{<t}, c). \tag{3}$$

Under this formulation, decoding proceeds across diagonal steps rather than individual tokens, reducing the number of sequential iterations from HW to $H + W - 1$. Therefore, the generation process changes from quadratic complexity over the 2D token grid to linear complexity in the spatial dimensions, enabling substantially more efficient autoregressive image synthesis.

4 Methodology

To avoid the prohibitive training costs of native parallel paradigms, our goal is to adapt a pre-trained visual AR model into a highly parallel two-way generator while aiming to retain its generative capabilities. Instead of retraining the backbone from scratch (He et al., 2025), we propose **FlashAR**, an elegant and lightweight post-training adaptation framework. This approach reuses the pre-trained causal decoder, introduces an additional vertical branch at a pre-final layer, and seamlessly fuses the predictions of orthogonal heads through a dynamic gating mechanism.

4.1 Pre-final Branching for Dual-head Decoding

Recent studies (Wu & Papyan, 2024; Skean et al., 2025) suggest that top-layer representations in deep autoregressive models can become increasingly specialized to the final prediction objective, losing general semantic abstractions. Moreover, the empirical observation illustrated in Figure 3 suggests that top-layer representations in a pre-trained raster-scan autoregressive model become specialized to the original left-to-right next-token prediction objective. Therefore, instead of introducing the new prediction branch at the final layer (He et al., 2025), we branch from a pre-final layer of the decoder, where the features remain semantically rich and are less tightly coupled to the original decoding direction.

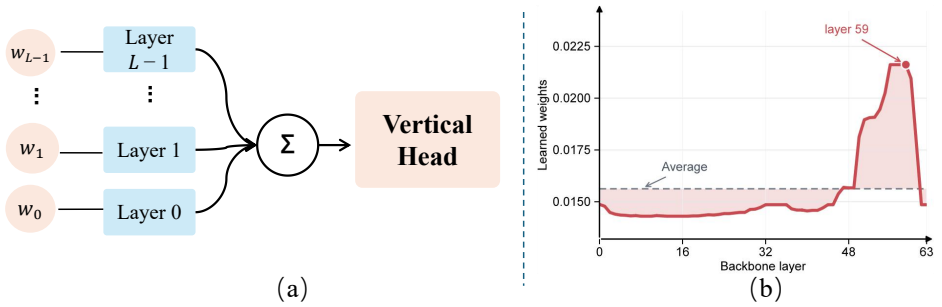


Figure 3: Analysis of linear probing experiments. (a) Schematic illustrating the aggregation of features from all Transformer layers using learnable weights before input to the vertical head. (b) The probe assigns greater weight to pre-final features than to the final-layer representation.

For the probing experiment in Figure 3, we freeze the backbone and jointly train the layer weights and vertical prediction head using the vertical next-token loss.

Let the pre-trained decoder backbone consist of L transformer layers. We denote the sequence of layers as $F = (F_0, F_1, \ldots, F_{L-1})$. Concretely, given newly generated token $y_{p,q}$, both branches share a unified backbone of depth m:

$$h_{p,q} = F_{0:m-1}(y_{p,q}), \tag{4}$$

From this shared intermediate state, we construct two distinct pathways:

$$z^H_{p,q} = Head^H(F_{m:L-1}(h_{p,q})), \tag{5}$$

$$z^V_{p,q} = Head^V(\widetilde{F}_{m:L-1}(h_{p,q})). \tag{6}$$

Here, the horizontal branch utilizes the original top $L - m$ layers $F_{m:L-1}$ in the pre-trained model, whereas the vertical branch employs $\widetilde{F}_{m:L-1}$, an independently trainable block initialized as a clone of the top $L - m$ layers. The original decoding head $Head^H$ is attached to the horizontal branch, serving for row-wise prediction. A complementary vertical decoding head $Head^V$ is attached to the vertical branch for column-wise prediction.

Compared to NAR (He et al., 2025), this branching design also improves runtime efficiency. Attaching a block after final layer would extend the critical-path depth. In contrast, branching at depth m allows the horizontal and vertical blocks to execute concurrently on top of the shared trunk, effectively maintaining the original critical-path depth.

4.2 Learnable Fusion Gate

The horizontal and vertical heads provide two complementary predictions for each image token. While simple averaging adopted by NAR (He et al., 2025) assumes an isotropic contribution from both horizontal and vertical contexts, natural images often exhibit significant anisotropy. For instance, when predicting a pixel along a sharp horizontal edge, the vertical predecessor may provide more reliable structural continuity than the horizontal one. A static average fails to capture such directional dependencies. Moreover, when the horizontal and vertical heads yield conflicting logit distributions, direct averaging acts as a low-pass filter in the probability space, which may lead to blurred textures or artifacts. We therefore fuse the two predictions with a learnable fusion gate, rather than simply averaging them.

Specifically, for $p > 0$ and $q > 0$, we compute a context-dependent gate from the two predecessor states:

$$g_{p,q} = \sigma\big(\mathrm{MLP}\big([\, h^H_{p,q-1};\, h^V_{p-1,q}\,]\big)\big)\,, \tag{7}$$

where $[\,\cdot;\cdot\,]$ denotes concatenation and σ is the sigmoid function. The fused logit is then given by

$$z_{p,q} = g_{p,q}\, z^H_{p,q-1} + (1 - g_{p,q})\, z^V_{p-1,q}. \tag{8}$$

Notably, for boundary positions, only the available directional prediction is used: the first row is decoded by horizontal logits, and the first column is decoded by vertical logits. The corner token $(0, 0)$ is predicted from the conditioning prefix.

4.3 Stabilized Adaptation and Training Objectives

Two-stage stabilized adaptation. To retain the generative capabilities of the pre-trained AR model while seamlessly adapting it for parallel decoding, we adopt a stabilized, two-stage post-training paradigm:

- *Stage 1: Branch Initialization.* In the initial phase, we completely freeze the pre-trained transformer backbone along with the original horizontal head. Only the newly initialized vertical head and the dynamic gating mechanism are set as trainable. This frozen-backbone strategy effectively prevents catastrophic forgetting of the pre-trained visual manifold and ensures highly stable, rapid convergence for the new components.
- *Stage 2: Joint Full Fine-Tuning.* Once the vertical pathway is sufficiently aligned, we unfreeze the entire network. The backbone, horizontal head, vertical head, and gating module are jointly fine-tuned. This stage harmonizes the orthogonal spatial representations and fully optimizes the network for 2D diagonal-parallel generation.

Training objectives. To optimize the model across these stages, we employ a comprehensive objective function comprising fusion and auxiliary losses. We supervise the final gated logits ($z_{p,q}$) via the standard Cross-Entropy (CE) loss to ensure target fidelity:

$$\mathcal{L}_{\text{fuse}} = \frac{1}{HW} \sum_{p=0}^{H-1} \sum_{q=0}^{W-1} \text{CE}(z_{p,q},\, y_{p,q})\,. \tag{9}$$

Furthermore, to ensure the standalone predictive capability of both the horizontal and vertical pathways under dynamic gating, we apply auxiliary CE losses to their independent predictions:

$$\begin{aligned} \mathcal{L}_H &= \frac{1}{H(W-1)} \sum_{p=0}^{H-1} \sum_{q=0}^{W-2} \text{CE}\big(z^H_{p,q},\, y_{p,q+1}\big)\,, \\ \mathcal{L}_V &= \frac{1}{(H-1)W} \sum_{p=0}^{H-2} \sum_{q=0}^{W-1} \text{CE}\big(z^V_{p,q},\, y_{p+1,q}\big)\,. \end{aligned} \tag{10}$$

Finally, the overall training objective is defined as: $\mathcal{L} = \mathcal{L}_{\text{fuse}} + \lambda_{\text{aux}}(\mathcal{L}_H + \mathcal{L}_V)$.

4.4 Inference-Time Parallelization with KV Cache

During inference, image generation proceeds iteratively along diagonals $\mathcal{D}_t$ in strict parallel, where logits for $\mathcal{D}_t$ are fused directly from the preceding diagonal $\mathcal{D}_{t-1}$. To translate this theoretical parallelism into inference speedups, we implement a hardware-aware inference pipeline. First, to enforce our 2D diagonal-step masking without the severe memory fragmentation caused by dense zero-padding, we integrate FlexAttention (Dong et al., 2024). This optimization dynamically compiles our highly sparse proximity mask on the fly, entirely bypassing the materialization of the full attention matrix. Second, after concurrently sampling $\mathcal{D}_t$, all tokens are appended to the branch-specific KV caches in a single batched operation. We similarly batch the conditional and unconditional forward passes for classifier-free guidance (Ho & Salimans, 2022), effectively amortizing kernel launch overheads and maximizing GPU arithmetic intensity.

5 Experiments

5.1 Experimental Setup

Training details. Our main experiments use LlamaGen (Sun et al., 2024) for ImageNet 256×256 class-conditional image generation, and Emu3.5-Image-34B (Cui et al., 2025) and GLM-Image (Zhipu AI, 2026) for text-to-image synthesis at 512×512 resolution. For LlamaGen, post-training is conducted for 25 epochs on the ImageNet dataset (Russakovsky et al., 2015) with a batch size of 512. For Emu3.5-Image-34B and GLM-Image, we curate a compact corpus of approximately 80K text-image pairs from OpenGPT-4o-Image (Chen et al., 2025b) and ShareGPT-4o-Image (Chen et al., 2025a), running the post-training stage for 50K steps.

Table 1: Quantitative evaluation on the ImageNet 256×256 benchmark

Model size	Method	Type	Training Epoch	FID↓	IS↑	P/R-F1↑	Steps	Throughput (img/s)
B (∼120M)	LlamaGen	From scratch	300	5.46	193.6	0.594	256	120.3
	PAR	From scratch	300	6.21	204.4	0.537	67	174.1
	NAR	From scratch	300	**4.65**	**212.3**	0.600	31	419.7
	BlockDiffusion	Post-training	75	5.91	176.2	0.589	64	186.3
	FlashAR	Post-training	25	4.68	208.3	**0.605**	31	**447.2**
L (∼360M)	LlamaGen	From scratch	300	3.80	248.3	0.639	256	47.2
	PAR	From scratch	300	4.32	189.4	0.576	67	93.8
	NAR	From scratch	300	**3.06**	263.9	0.641	31	197.3
	VAR	From scratch	200	3.30	274.4	0.634	10	134.8
	BlockDiffusion	Post-training	75	4.55	243.5	0.645	64	103.2
	FlashAR	Post-training	25	3.16	**289.0**	**0.652**	31	**224.7**
XL (∼700M)	LlamaGen	From scratch	300	3.39	227.1	0.648	256	23.7
	PAR	From scratch	300	3.50	234.4	0.619	67	53.9
	NAR	From scratch	300	**2.70**	277.5	**0.676**	31	97.4
	BlockDiffusion	Post-training	75	4.13	258.6	0.654	64	41.7
	FlashAR	Post-training	25	2.94	**293.7**	0.672	31	**109.3**
XXL (∼1.4B)	LlamaGen	From scratch	300	3.09	253.6	0.647	256	14.1
	PAR	From scratch	300	3.20	288.3	0.632	67	33.9
	NAR	From scratch	300	**2.58**	**293.5**	0.673	31	57.8
	BlockDiffusion	Post-training	75	3.78	264.9	0.652	64	26.8
	FlashAR	Post-training	25	2.79	289.4	**0.690**	31	**63.4**

As detailed in Section 4.3, we adopt a two-stage training schedule. Stage 1 freezes the backbone during the first 20% of the training steps to stabilize the vertical head, whereas stage 2 unlocks the full network and scales the backbone learning rate by a factor of 0.2 relative to the head. Both stages use a base learning rate of 2×10^{-5} with cosine decay. Loss coefficients are $\lambda_{\text{aux}} = 0.05$. Classifier-free guidance (cfg) is set to 2.0 for LlamaGen and 5.0 for Emu3.5 by default. All experiments are run on 8 NVIDIA H20 GPUs with bf16 precision.

Baselines and metrics. We compare FlashAR against three categories of baselines. (i) Raster-scan AR: the original LlamaGen and Emu3.5-Image checkpoints with sequential raster-scan decoding. (ii) Parallel AR trained from scratch: VAR (Tian et al., 2024), PAR (Wang et al., 2025b), and NAR (He et al., 2025), which adopt distinct parallelization strategies but all require training from scratch under modified generation objectives. (iii) Post-training adaptation: Block Diffusion, which is adopted by Emu3.5 (Cui et al., 2025) and enables parallel generation via discrete diffusion adaptation. For LlamaGen, we report FID (Heusel et al., 2017), Inception Score (IS) (Salimans et al., 2016), Precision and Recall (Kynkäänniemi et al., 2019), and sFID as generative quality metrics. For text-to-image generation, we report GenEval (Ghosh et al., 2023) on Emu3.5-Image. On Emu3.5-Image, we additionally use DPG-Bench (Hu et al., 2024), T2I-CompBench (Huang et al., 2023), and PickScore (Kirstain et al., 2023) to assess dense-prompt alignment, fine-grained composition, and predicted human preference, respectively. Inference latency is measured on a single NVIDIA H20-96G GPU with batch size 1, averaged over 100 samples.

5.2 Main Results

Results on LlamaGen. Table 1 presents the class-conditional image generation results on ImageNet at 256×256 resolution. Among existing post-training methods, FlashAR significantly outperforms BlockDiffusion in both quality and efficiency. Specifically, at the L scale, FlashAR achieves a superior FID (3.16 vs. 4.55) using only 25 adaptation epochs—one-third of BlockDiffusion's training budget. More notably, FlashAR-L obtains an IS that surpasses NAR-L (289.0 vs. 263.9), a model trained entirely from scratch, despite FlashAR requiring only lightweight post-training on a pre-trained raster-AR backbone. Additionally, FlashAR-B achieves a throughput of **447.21** images/s, outperforming even NAR-B (419.7 images/s); this efficiency gain is attributable to the inference-time optimizations detailed in Section 4.4. Overall, these results demonstrate that the proposed post-training approach yields both competitive generation quality and superior inference efficiency across various model scales.

Results on Emu3.5-Image. We further evaluate FlashAR on Emu3.5-Image-34B (Cui et al., 2025) for 512×512 text-to-image generation. After 50K post-training steps using 80M tokens—approximately 0.053% of the backbone's reported pre-training tokens—FlashAR reduces the number of sequential decoding steps from 1024 to 63. As shown in Table 2, this reduces latency from 103.57 to 7.95 seconds per image, yielding a $13.03\times$ speedup over the fastest evaluated AR baseline with batched CFG enabled for both methods.

FlashAR also retains competitive generation quality. As shown in Table 3, it achieves an overall GenEval score of 80.16, compared with 80.49 for the original model and 74.23 for our BlockDiffusion reproduction. The detailed results show improved spatial relations but weaker color-attribute binding. Table 4 reports additional text-to-image evaluations: DPG-Bench and PickScore decrease slightly, while T2I-CompBench improves in four dimensions with modest decreases in shape and texture. We also apply FlashAR to accelerate the autoregressive component of GLM-Image, with detailed results provided in Appendix B.

Table 2: Inference efficiency on Emu3.5-Image (512×512, CFG$= 5.0$). Speedups are relative to the fastest evaluated AR baseline; both it and FlashAR use batched CFG.

Method	Attention backend	Batched CFG	Latency (s)	Speedup
Original AR	FlashAttention-2	No	189.36	$0.55\times$
Original AR	FlexAttention	Yes	121.83	$0.85\times$
Original AR	FlashAttention-2	Yes	103.57	$1.00\times$
FlashAR	FlexAttention	Yes	7.95	$13.03\times$

Table 3: GenEval scores on Emu3.5-Image-34B (cfg = 5.0, 512×512).

Method	**Overall**	**Single Obj**	**Two Obj**	**Counting**	**Colors**	**Position**	**Color_attr**
Emu3.5-Image	80.49	100.00	94.95	53.75	90.96	73.00	70.25
BlockDiffusion	74.23	96.88	88.89	47.50	85.64	68.00	58.44
FlashAR	80.16	98.75	91.92	53.75	92.55	80.00	64.00

Table 4: Text-to-image generation results on Emu3.5-Image.

Method	GenEval ↑	DPG-Bench ↑	PickScore ↑	T2I-CompBench					
				Non-spatial ↑	Complex ↑	Color ↑	Shape ↑	Texture ↑	Spatial ↑
Original AR	80.49	86.74	0.2309	0.31	0.30	0.81	0.63	0.75	0.41
FlashAR	80.16	86.12	0.2262	0.32	0.32	0.82	0.61	0.73	0.42

5.3 Ablation Study

Post-training efficiency comparison with block diffusion. To evaluate the convergence efficiency of our approach, we plot the FID trajectories across training epochs in Figure 4(a). As illustrated, FlashAR demonstrates significantly faster convergence and superior generation quality compared to the BlockDiffusion baseline. Remarkably, after only 5 epochs of adaptation, the two-stage FlashAR achieves an FID of 3.98, which already comfortably outperforms the final performance of BlockDiffusion at 25 epochs (4.55). Furthermore, the results validate the efficacy of our proposed two-stage fine-tuning schedule over the conventional one-stage baseline, where the two-stage schedule ultimately reaches a superior FID of 3.16 compared to 3.27 for the one-stage baseline at epoch 25.

Effect of branching depth. Table 5 evaluates the impact of the branching depth m. Excessive branching depth limits the expressive capacity of the vertical pathway, whereas an insufficient depth reduces trunk sharing and increases parameter overhead. We find that pre-final branching ($m = 23$ of $L = 24$ layers) yields the optimal trade-off. This observation is consistent with the component ablation results presented in Figure 4(b) under a different cfg scale. Specifically, introducing the vertical branch alone improves the FID over naive adaptation (4.16 vs. 4.36), validating our design choice to branch from pre-final layers rather than relying solely on the final-layer representation.

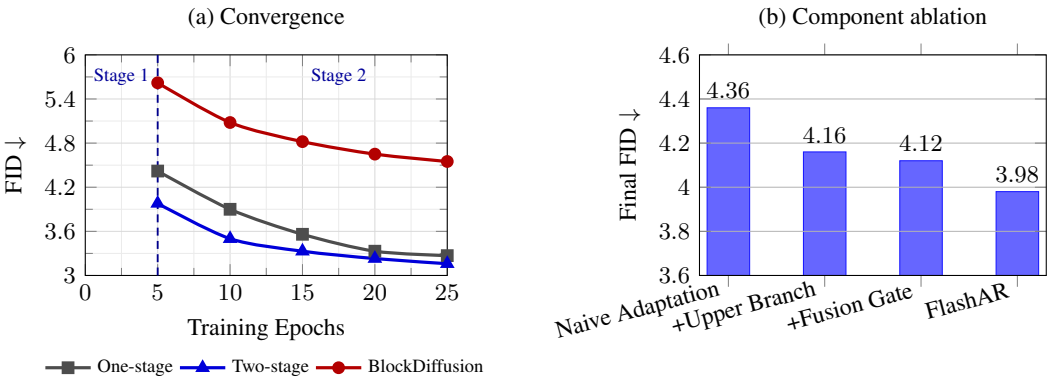


Figure 4: Ablation studies on LlamaGen-L. **(a)** FID convergence trajectories across training epochs at CFG= 2.0. **(b)** Final FID comparisons across component variants at CFG= 1.5.

Table 5: Effect of branching depth m on LlamaGen-L ($L = 24$, cfg= 2.0).

Branch Depth	FID ↓	sFID ↓	IS ↑	Prec. ↑	Recall ↑
$m = 24$	3.29	6.41	288.0	**0.835**	0.529
$m = 23$ (ours)	**3.16**	6.34	**289.0**	0.834	**0.535**
$m = 21$	3.20	**6.32**	285.9	0.833	0.518

Effect of fusion gate design. Table 6 compares various strategies for combining horizontal and vertical predictions. Fixed averaging is consistently inferior to our learnable fusion gate in terms of FID, sFID, and IS, indicating that the relative importance of the two directional cues should be determined adaptively rather than prescribed globally. The component ablation in Figure 4(b) under a different CFG scale further demonstrates the efficacy of the learnable fusion gate, which reduces the FID from 4.36 to 4.12.

Table 6: Ablation on fusion strategies (cfg=1.5).

Fusion Strategy	FID ↓	sFID ↓	IS ↑	Prec. ↑	Recall ↑
Average ($g = 0.5$)	4.36	6.32	187.8	0.756	**0.604**
Learnable Gate (ours)	**4.12**	**6.30**	**191.4**	**0.765**	0.596

6 Conclusion

In this paper, we have proposed FlashAR, a lightweight post-training adaptation framework that efficiently transforms a pre-trained raster-scan autoregressive model into a highly parallel generator. To bypass the inherent horizontal bias of the pre-trained model and mitigate context conflicts, we have branched the vertical pathway from a pre-final layer and have integrated a learnable fusion gate to dynamically combine directional predictions at each target position. Furthermore, we have designed a two-stage post-training pipeline that minimizes adaptation overhead through vertical branch initialization followed by joint fine-tuning. On the deployment front, we have implemented a hardware-aware inference pipeline leveraging FlexAttention and batched KV-cache updates. Extensive experiments on LlamaGen and Emu3.5 have demonstrated a 13.03× speedup on Emu3.5-Image using approximately 0.05% of its pre-training tokens, while retaining competitive generation quality.

Limitations and future work. FlashAR has two primary limitations: selecting the pre-final branching depth requires empirical search, and parallel generation imposes conditional independence among tokens on the same anti-diagonal, preventing intra-diagonal interaction. Future work will explore automated branch placement, lightweight intra-step communication (e.g., iterative refinement), and extensions to 3D spatio-temporal tasks such as autoregressive video generation.

Reproducibility Statement

To facilitate reproducibility, Section 4 describes the dual-head architecture, learnable fusion gate, training objectives, two-stage adaptation schedule, and inference implementation. Section 5.1 reports the pre-trained backbones, training data sources, adaptation budgets, learning-rate schedule, loss coefficients, classifier-free guidance settings, hardware, and numerical precision. It also specifies the evaluation benchmarks and metrics. Inference latency is measured on a single NVIDIA H20-96G GPU with batch size 1 and averaged over 100 samples; Table 2 additionally identifies the attention backend and batched guidance setting for each Emu3.5 timing comparison. Section 5.3 provides controlled ablations of the adaptation schedule, branching depth, and fusion strategy.

AI use statement

We used AI tools for language polishing, grammar checking, and code debugging. All research design, technical content, data analysis, interpretations, and conclusions have been independently developed, reviewed, and verified by the authors, who take full responsibility for the accuracy and integrity of the work.

References

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.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

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*, 2025a.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pp. 1691–1703. PMLR, 2020.

Zhihong Chen, Xuehai Bai, Yang Shi, Chaoyou Fu, Huanyu Zhang, Haotian Wang, Xiaoyan Sun, Zhang Zhang, Liang Wang, Yuanxing Zhang, et al. OpenGPT-4o-Image: A comprehensive dataset for advanced image generation and editing. *arXiv preprint arXiv:2509.24900*, 2025b.

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

Haoge Deng, Ting Pan, Fan Zhang, Yang Liu, Zhuoyan Luo, Yufeng Cui, Wenxuan Wang, Chunhua Shen, Shiguang Shan, Zhaoxiang Zhang, et al. Uniform discrete diffusion with metric path for video generation. *arXiv preprint arXiv:2510.24717*, 2025.

Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. Flex Attention: A programming model for generating optimized attention kernels. *arXiv preprint arXiv:2412.05496*, 2024.

Zicheng Duan, Jiatong Xia, Zeyu Zhang, Wenbo Zhang, Gengze Zhou, Chenhui Gou, Yefei He, Feng Chen, Xinyu Zhang, and Lingqiao Liu. LiveWorld: Simulating out-of-sight dynamics in generative video world models. *arXiv preprint arXiv:2603.07145*, 2026.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12873–12883, 2021.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37: 133345–133385, 2024.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

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.

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.

Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15733–15744, 2025.

Yefei He, Yuanyu He, Shaoxuan He, Feng Chen, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. Neighboring autoregressive modeling for efficient visual generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19000–19010, 2025.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 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.

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2I-CompBench: A comprehensive benchmark for open-world compositional text-to-image generation. *arXiv preprint arXiv:2307.06350*, 2023.

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.

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. *arXiv preprint arXiv:2305.01569*, 2023.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/0234c510bc6d908b28c70ff313743079-Paper.pdf.

Haopeng Li, Jinyue Yang, Guoqi Li, and Huan Wang. Autoregressive image generation with randomized parallel decoding. *arXiv preprint arXiv:2503.10568*, 2025.

Dongyang Liu, Shitian Zhao, Le Zhuo, Weifeng Lin, Yi Xin, Xinyue Li, Qi Qin, Yu Qiao, Hongsheng Li, and Peng Gao. Lumina-mGPT: Illuminate flexible photorealistic text-to-image generation with multimodal generative pretraining. *arXiv preprint arXiv:2408.02657*, 2024.

NextStep Team, Chunrui Han, Guopeng Li, Jingwei Wu, Quan Sun, Yan Cai, Yuang Peng, Zheng Ge, Deyu Zhou, Haomiao Tang, et al. NextStep-1: Toward autoregressive image generation with continuous tokens at scale. *arXiv preprint arXiv:2508.10711*, 2025.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. *Advances in Neural Information Processing Systems*, 29, 2016.

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.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*, 2025.

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*, 2024.

Haotian Tang, Yecheng Wu, Shang Yang, Enze Xie, Junsong Chen, Junyu Chen, Zhuoyang Zhang, Han Cai, Yao Lu, and Song Han. HART: Efficient visual generation with hybrid autoregressive transformer. *arXiv preprint arXiv:2410.10812*, 2024.

Yao Teng, Han Shi, Xian Liu, Xuefei Ning, Guohao Dai, Yu Wang, Zhenguo Li, and Xihui Liu. Accelerating auto-regressive text-to-image generation with training-free speculative Jacobi decoding. *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. *Advances in Neural Information Processing Systems*, 37:84839–84865, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with PixelCNN decoders. *Advances in Neural Information Processing Systems*, 29, 2016.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Junke Wang, Zhi Tian, Xun Wang, Xinyu Zhang, Weilin Huang, Zuxuan Wu, and Yu-Gang Jiang. SimpleAR: Pushing the frontier of autoregressive visual generation through pretraining, SFT, and RL. *arXiv preprint arXiv:2504.11455*, 2025a.

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.

Yuqing Wang, Shuhuai Ren, Zhijie Lin, Yujin Han, Haoyuan Guo, Zhenheng Yang, Difan Zou, Jiashi Feng, and Xihui Liu. Parallelized autoregressive visual generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12955–12965, 2025b.

Zili Wang, Robert Zhang, Kun Ding, Qi Yang, Fei Li, and Shiming Xiang. Continuous speculative decoding for autoregressive image generation. *arXiv preprint arXiv:2411.11925*, 2024b.

Robert Wu and Vardan Papyan. Linguistic collapse: Neural collapse in (large) language models. *Advances in Neural Information Processing Systems*, 37:137432–137473, 2024.

Yi Xin, Juncheng Yan, Qi Qin, Zhen Li, Dongyang Liu, Shicheng Li, Victor Shea-Jay Huang, Yupeng Zhou, Renrui Zhang, Le Zhuo, et al. Lumina-mGPT 2.0: Stand-alone autoregressive image modeling. *arXiv preprint arXiv:2507.17801*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

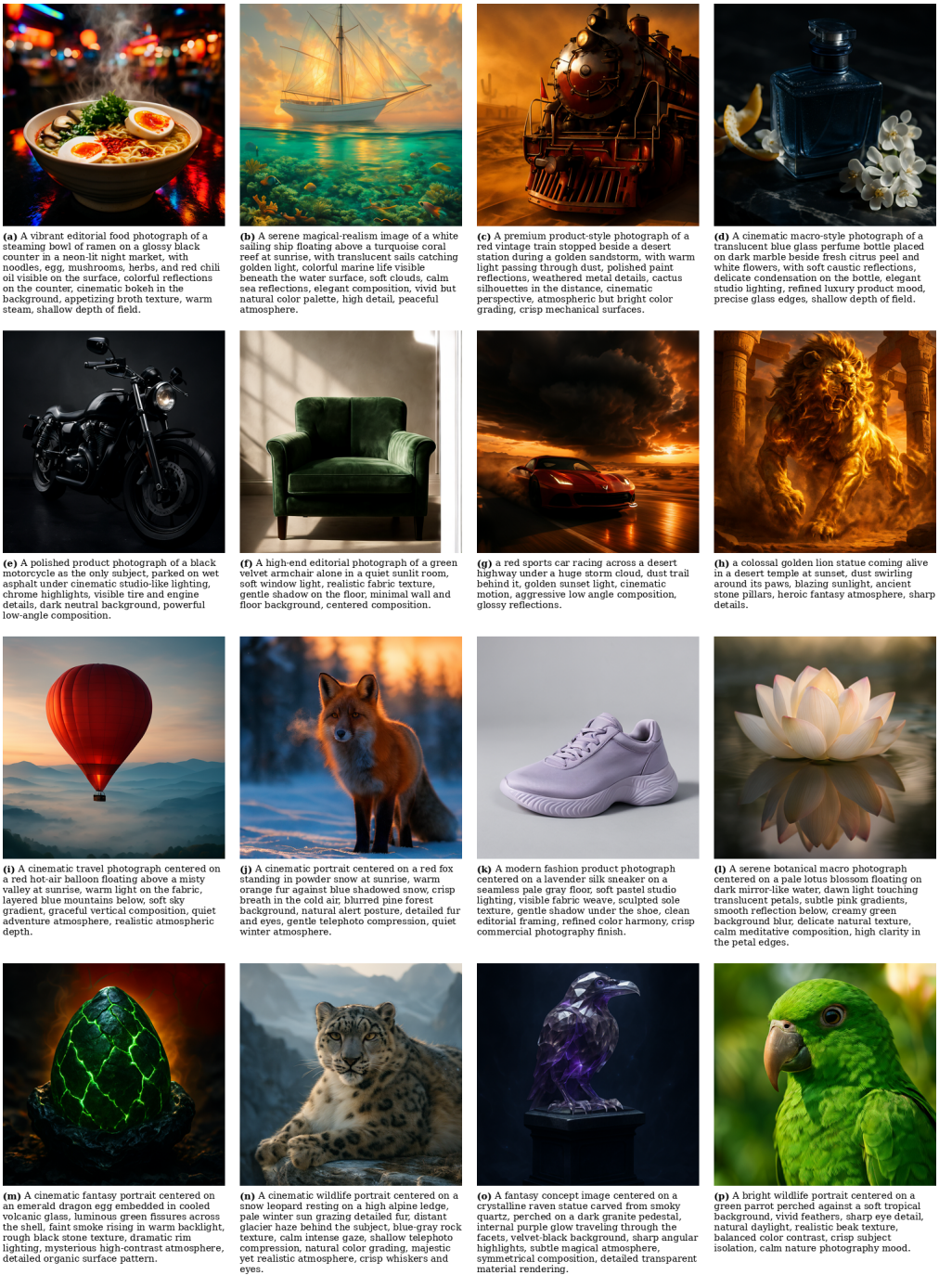
Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022.

Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18431–18441, 2025.

Zhuoyang Zhang, Luke J Huang, Chengyue Wu, Shang Yang, Kelly Peng, Yao Lu, and Song Han. Locality-aware parallel decoding for efficient autoregressive image generation. *arXiv preprint arXiv:2507.01957*, 2025.

Zhipu AI. GLM-Image: Auto-regressive for dense-knowledge and high-fidelity image generation. https://z.ai/blog/glm-image, January 2026. Accessed: 2026-04-30.

A VISUALIZATION



(a) A vibrant editorial food photograph of a steaming bowl of ramen on a glossy black counter in a neon-lit night market, with noodles, egg, mushrooms, herbs, and red chili oil visible on the surface, colorful reflections on the counter, cinematic bokeh in the background, appetizing broth texture, warm steam, shallow depth of field.

(b) A serene magical-realism image of a white sailing ship floating above a turquoise coral reef at sunrise, with translucent sails catching golden light, colorful marine life visible beneath the water surface, soft clouds, calm sea reflections, elegant composition, vivid but natural color palette, high detail, peaceful atmosphere.

(c) A premium product-style photograph of a red vintage train stopped beside a desert station during a golden sandstorm, with warm light passing through dust, polished paint reflections, weathered metal details, cactus silhouettes in the distance, cinematic perspective, atmospheric but bright color grading, crisp mechanical surfaces.

(d) A cinematic macro-style photograph of a translucent blue glass perfume bottle placed on dark marble beside fresh citrus peel and white flowers, with soft caustic reflections, delicate condensation on the bottle, elegant studio lighting, refined luxury product mood, precise glass edges, shallow depth of field.

(e) A polished product photograph of a black motorcycle as the only subject, parked on wet asphalt under cinematic studio-like lighting, chrome highlights, visible tire and engine details, dark neutral background, powerful low-angle composition.

(f) A high-end editorial photograph of a green velvet armchair alone in a quiet sunlit room, soft window light, realistic fabric texture, gentle shadow on the floor, minimal wall and floor background, centered composition.

(g) a red sports car racing across a desert highway under a huge storm cloud, dust trail behind it, golden sunset light, cinematic motion, aggressive low angle composition, glossy reflections.

(h) a colossal golden lion statue coming alive in a desert temple at sunset, dust swirling around its paws, blazing sunlight, ancient stone pillars, heroic fantasy atmosphere, sharp details.

(i) A cinematic travel photograph centered on a red hot-air balloon floating above a misty valley at sunrise, warm light on the fabric, layered blue mountains below, soft sky gradient, graceful vertical composition, quiet adventure atmosphere, realistic atmospheric depth.

(j) A cinematic portrait centered on a red fox standing in powder snow at sunrise, warm orange fur against blue shadowed snow, crisp breath in the cold air, blurred pine forest background, natural alert posture, detailed fur and eyes, gentle telephoto compression, quiet winter atmosphere.

(k) A modern fashion product photograph centered on a lavender silk sneaker on a seamless pale gray floor, soft pastel studio lighting, visible fabric weave, sculpted sole texture, gentle shadow under the shoe, clean editorial framing, refined color harmony, crisp commercial photography finish.

(l) A serene botanical macro photograph centered on a pale lotus blossom floating on dark mirror-like water, dawn light touching translucent petals, subtle pink gradients, smooth reflection below, creamy green background blur, delicate natural texture, calm meditative composition, high clarity in the petal edges.

(m) A cinematic fantasy portrait centered on an emerald dragon egg embedded in cooled volcanic glass, luminous green fissures across the shell, faint smoke rising in warm backlight, rough black stone texture, dramatic rim lighting, mysterious high-contrast atmosphere, detailed organic surface pattern.

(n) A cinematic wildlife portrait centered on a snow leopard resting on a high alpine ledge, pale winter sun grazing detailed fur, distant glacier haze behind the subject, blue-gray rock texture, calm intense gaze, shallow telephoto compression, natural color grading, majestic yet realistic atmosphere, crisp whiskers and eyes.

(o) A fantasy concept image centered on a crystalline raven statue carved from smoky quartz, perched on a dark granite pedestal, internal purple glow traveling through the facets, velvet-black background, sharp angular highlights, subtle magical atmosphere, symmetrical composition, detailed transparent material rendering.

(p) A bright wildlife portrait centered on a green parrot perched against a soft tropical background, vivid feathers, sharp eye detail, natural daylight, realistic beak texture, balanced color contrast, crisp subject isolation, calm nature photography mood.

Figure 5: **Complex text-guided image generation samples** by Emu3.5-Image-FlashAR



class id 387, lesser panda



class id 90, lorikeet



class id 250, Siberian husky



class id 933, cheeseburger



class id 437, beacon



class id 979, valley



class id 985, daisy

Figure 6: **Class-conditional image generation samples** produced by FlashAR-XXL on ImageNet 256×256

B Additional Results on GLM-Image

Beyond Emu3.5-Image, we apply FlashAR to accelerate the autoregressive component of GLM-Image (Zhipu AI, 2026). As shown in Table 7, we compare against the original AR implementation and evaluate text-to-image generation quality with GenEval (Ghosh et al., 2023). With 40M post-training tokens, FlashAR reduces latency from 110.17 to 8.92 seconds per image (12.35×), while GenEval decreases from 70.74 to 68.54.

Table 7: Text-to-image generation results on GLM-Image. Speedup is measured relative to the original AR implementation used in this experiment.

Method	Latency (s) ↓	Speedup	GenEval ↑						
			Overall	Single Obj	Two Obj	Counting	Colors	Position	Color Attr.
Original AR	110.17	1.00×	70.74	95.14	81.73	45.29	88.16	64.82	49.35
FlashAR	8.92	12.35×	68.54	94.62	79.84	44.81	87.42	61.35	43.19