Drift-AR: Single-Step Visual Autoregressive Generation via Anti-Symmetric Drifting

Zhen Zou[1], Xiaoxiao Ma[1], Mingde Yao[2], Jie Huang[3†], Linjiang Huang[4], and Feng Zhao[1⋆]

[1] MoE Key Lab of BIPC, University of Science and Technology of China
[2] MMLab, The Chinese University of Hong Kong
[3] JD Explore Academy
[4] Beihang University
zouzhen@mail.ustc.edu.cn, fzhao956@ustc.edu.cn

Abstract. Autoregressive (AR)-Diffusion hybrid paradigms combine AR's structured semantic modeling with diffusion's high-fidelity synthesis, yet suffer from a dual speed bottleneck: the sequential AR stage and the iterative multi-step denoising of the diffusion vision decode stage. Existing methods address each in isolation without a unified principle design. We observe that the per-position *prediction entropy* of continuous-space AR models naturally encodes spatially varying generation uncertainty, which simultaneously governing draft prediction quality in the AR stage and reflecting the corrective effort required by vision decoding stage, which is not fully explored before. Since entropy is inherently tied to both bottlenecks, it serves as a natural unifying signal for joint acceleration. In this work, we propose **Drift-AR**, which leverages entropy signal to accelerate both stages: 1) for AR acceleration, we introduce Entropy-Informed Speculative Decoding that align draft–target entropy distributions via a causal-normalized entropy loss, resolving the entropy mismatch that causes excessive draft rejection; 2) for visual decoder acceleration, we reinterpret entropy as the *physical variance* of the initial state for an anti-symmetric drifting field—high-entropy positions activate stronger drift toward the data manifold while low-entropy positions yield vanishing drift—enabling single-step (1-NFE) decoding without iterative denoising or distillation. Moreover, both stages share the same entropy signal, which is computed once with no extra cost. Experiments on MAR, TransDiff, and NextStep-1 demonstrate 3.8–5.5× speedup with genuine 1-NFE decoding, matching or surpassing original quality. Code is available at https://github.com/aSleepyTree/Drift-AR.

Keywords: Autoregressive generation · Drifting models · Single-step generation

1 Introduction

Autoregressive (AR) models have achieved remarkable success in large language models (LLMs) [1, 2, 33], inspiring extensions to visual generation [19, 30]. To

† Project leader.
⋆ Corresponding author.

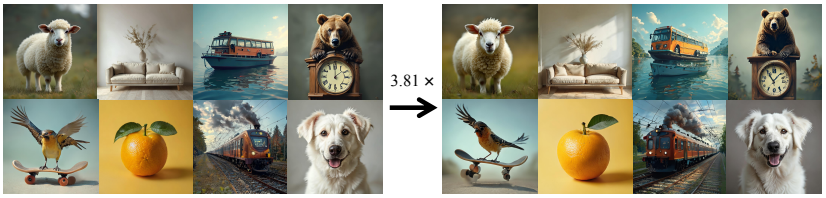

Fig. 1: Qualitative generation comparison between the vanilla NextStep-1 [31] and our method on GenEval [8].

overcome the quality ceiling imposed by discrete tokenization [6, 34], recent hybrid AR-Diffusion paradigms [13, 22, 31, 42] shift generation to continuous latent spaces with two stages, combining AR stage's structured semantic modeling with diffusion decoder stage's high-fidelity synthesis [10, 27, 29]. These hybrids have achieved impressive results, balancing semantic coherence and visual realism without the information loss inherent to vector quantization.

However, the two-stage design introduces a *dual speed bottleneck*. In this hybrid inference paradigm, the AR transformer generates tokens sequentially. Each generated AR latent is then decoded by the diffusion decoder through iterative multi-step denoising. Existing acceleration methods treat these two bottlenecks independently: 1) speculative decoding [14, 15] for the AR acceleration and, 2) typical distillation [38, 39] for the visual decoder acceleration. This isolated treatment lacks a unified guiding principle and leaves substantial room for their joint acceleration. Moreover, distillation methods such as consistency or distribution-matching distillation (CD/DMD) [28, 38, 39] remain tied to the multi-step diffusion paradigm: although they reduce the number of denoising steps, iterative inference is still required.

We start our work to accelerate the AR-Diffusion hybrids generation by making a key observation: unlike LLMs generation, real-world images exhibit highly non-uniform information distribution, leading to heterogeneous generation uncertainty across spatial positions. In continuous-space AR models, the *per-position predicted entropy* naturally captures this heterogeneity, providing a signal with deep implications for both AR acceleration and diffusion visual decoding acceleration, which is not fully explored by existing methods. Specifically, redundant regions (*e.g.*, sky, blank walls) tend to induce low entropy, whereas complex structures (textures, object boundaries) produce high entropy. Crucially, this entropy carries a *dual meaning* for the dual speed bottleneck: (1) it reflects the reliability of draft predictions in speculative decoding: entropy mismatch between small draft models and target large models often leads to excessive rejection (see Fig. 2(a)); and (2) it encodes the local generation difficulty at each spatial position, indicating how much "corrective effort" the diffusion-derived visual decoder should apply. We therefore argue that entropy forms a *natural bridge* that connects AR acceleration and diffusion visual decoding acceleration within a unified framework.

Based on the above observation and analysis, we first address the AR acceleration bottleneck via **Entropy-Informed Speculative Decoding** (see left in Fig. 3). Directly applying speculative decoding approach (e.g., EAGLE [15]) to continuous-space AR yields poor acceptance rates, as smaller draft models tend to produce low-entropy, overconfident features that deviate from the large target's entropy distribution (see Fig. 2(a)). This mismatch arises from the low information density of visual data, where the limited capacity of the draft model causes predictions to collapse toward dominant modes. To mitigate this issue, we adapt EAGLE to continuous latent spaces with two key modifications: (i) a continuous-feature regression loss that replaces the discrete classification objective that matches the vision AR model's characteristics, and (ii) a causal-normalized entropy loss that explicitly aligns the entropy distributions of the draft and target models, thereby improving acceptance rates of speculative decoding in AR acceleration.

We then address the visual decoding bottleneck by going beyond entropy as a simple confidence signal (see middle and right in Fig. 3). We replace the iterative diffusion-derived decoder with an **anti-symmetric drifting field** [5] and reinterpret the AR prediction entropy as the *physical variance* of the drifting process's initial distribution. This leads to an **Entropy-Parameterized Prior**: at each spatial position, the AR feature serves as the prior mean, while the entropy-derived variance determines the spread of the initial state. Empirically, this formulation is well supported. As shown in Fig. 2(c)(d), per-position AR prediction error strongly correlates with entropy (Pearson r=0.64), indicating that high-entropy regions correspond to locations where AR predictions deviate most from the ground truth and therefore require stronger correction during decoding. The formulation is physically consistent with the anti-symmetric drifting field ($V_{p,q}{=}{-}V_{q,p}$): high-entropy positions induce larger variance and thus stronger drift toward the data manifold, while low-entropy positions yield near-zero variance where the drift naturally vanishes ($q{\approx}p \Rightarrow V{\to}0$). This enables principled single-step generation for visual decoding stage without iterative refinement.

The two components are naturally unified through the shared entropy signal: the per-position entropy $\mathcal{E}^{(r)}_{\text{entropy}}$ computed once during speculative decoding is directly reused as the variance parameter for the visual decoder's initial distribution, requiring no additional computation or separate uncertainty estimation. Therefore, the proposed framework is end-to-end training, which is enabled by an annealed schedule that first stabilizes the AR entropy, then optimizes the drifting field over a fixed prior. Unlike CD/DMD-based acceleration, our framework eliminates the need for a multi-step teacher and avoids distillation instability, achieving 1-NFE visual decoding from first principles.

Our contributions are as follows:

- We identify the *dual role* of prediction entropy in continuous-space AR-Diffusion hybrids, which governs both draft quality in speculative decoding for AR stage and local generation difficulty for visual decoding stage, and thus identify entropy signal to unify the two stage's joint acceleration.

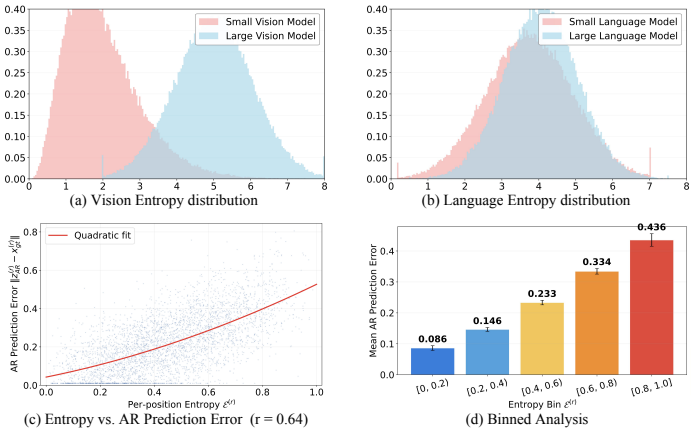

Fig. 2: Entropy as a diagnostic signal for AR-Diffusion hybrids. (a) Vision AR entropy: the draft model (red) concentrates at low entropy while the target model (blue) spans higher values, revealing severe *entropy mismatch*. (b) Language AR entropy: large and small models overlap substantially, explaining why speculative decoding succeeds in LLMs, which cannot be directly applied for vision AR models. (c) Per-position AR prediction error $\|z_{AR}^{(r)} - x_{gt}^{(r)}\|$ vs. entropy $\mathcal{E}^{(r)}$ (Pearson r=0.64): higher entropy correlates with larger prediction error, confirming that entropy encodes local generation difficulty. (d) Binned analysis: mean AR error increases monotonically with entropy, motivating entropy-parameterized variance for the drifting decoder.

- We introduce Drift-AR, which couples entropy-informed speculative decoding for AR stage acceleration, along with an entropy-parameterized anti-symmetric drifting field for visual decoding stage acceleration, achieving genuine single-step (1-NFE) visual generation without iterative denoising or distillation.
- Extensive experiments on MAR, TransDiff, and NextStep-1 demonstrate 3.8–5.5× speedup while matching or surpassing original generation quality and diversity.

2 Related Work

2.1 AR-Diffusion Hybrid Paradigm for Generation

The AR-Diffusion hybrid bridges AR's structured semantic modeling and diffusion's high-fidelity synthesis. MAR [13] introduced a diffusion-inspired denoising loss on continuous latent features, avoiding discrete vector quantization while preserving AR's sequential prediction. TransDiff [42] achieved end-to-end integration by coupling an AR transformer with a DiT-based diffusion decoder, mitigating AR's $O(n^2)$ complexity. FlowAR [22] replaced diffusion with flow matching [18,20] and added Spatial-adaLN for feature alignment, while NextStep-1 [31] unified AR and flow matching in a single transformer. These models share core

principles: AR supplies hierarchical semantic guidance, diffusion/flow matching handles high-dimensional visual synthesis, and tight integration via joint training or adaptive feature injection preserves guidance–synthesis alignment.

2.2 Acceleration of Generative Models

Acceleration techniques for generative models target the distinct bottlenecks of AR (sequential latency) and diffusion (iterative denoising) [3, 4, 7, 21, 25, 26, 32, 35, 40, 41]. For diffusion, distillation-based methods such as DMD [39] and DMD2 [38] align student–teacher distributions to reduce denoising to one or few steps, while SDXL-Lightning [17] combines progressive distillation [24] with adversarial loss. Other works train few-step or single-step generators directly [5]. However, synergistic acceleration for AR-Diffusion hybrids remains largely unaddressed, with current methods optimizing each component in isolation.

3 Preliminary

3.1 AR-Diffusion Hybrid Paradigm

As an AR-Diffusion hybrid model, MAR [13] operates directly on continuous latents $x \in \mathbb{R}^d$ with a diffusion-inspired objective:

$$\mathcal{L}_{MAR} = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I),t\sim\text{Uniform}(0,1)}\left[\|\epsilon - \epsilon_\theta(x_t \mid t, z_{AR})\|^2\right], \tag{1}$$

where $x_t = \sqrt{\bar{\alpha}_t}x + \sqrt{1-\bar{\alpha}_t}\epsilon$ is the noisy latent, z_{AR} denotes AR's predicted feature, and ϵ_θ represents the diffusion noise estimator.

TransDiff [42] further enables end-to-end integration: its AR transformer models sequential semantic features as $z = \text{ART}(x_{1:i-1}, c)$, where c is class/text condition and $x_{1:i-1}$ are preceding latents, ART is the AR diffusion transformer. A DiT-based diffusion decoder Diff_θ then reconstructs images via: $x_0 = \text{Diff}_\theta(x_t \mid t, z)$. Although TransDiff is capable of generating all tokens in one step, we treat it as a multi-step method for improved performance.

3.2 Acceleration

Speculative decoding accelerates AR generation via a "draft–verify" mechanism: a lightweight draft model proposes predictions in parallel, which a stronger target model verifies. EAGLE [15] and its successors [14, 16] are the state-of-the-art for *discrete-token* NLP models, employing feature regression and classification objectives:

$$\begin{aligned}\mathcal{L}_{\text{reg}} &= \text{Smooth L1}(f_{j+1}, \text{Draft Model}(T_{2:j+1}, F_{1:j})),\\ \mathcal{L}_{\text{cls}} &= \text{Cross-Entropy}(p_{j+2}, \hat{p}_{j+2}),\end{aligned} \tag{2}$$

where $F_{1:j}$ denotes draft model's current feature sequence, $\hat{f}_{j+1}$ the target model's next-step feature, $T_{2:j+1}$ the output tokens, and $p/\hat{p}$ the corresponding discrete probability distributions. Adapting EAGLE to the continuous-space AR-Diffusion setting requires non-trivial modifications, as we discuss in Sec. 4.

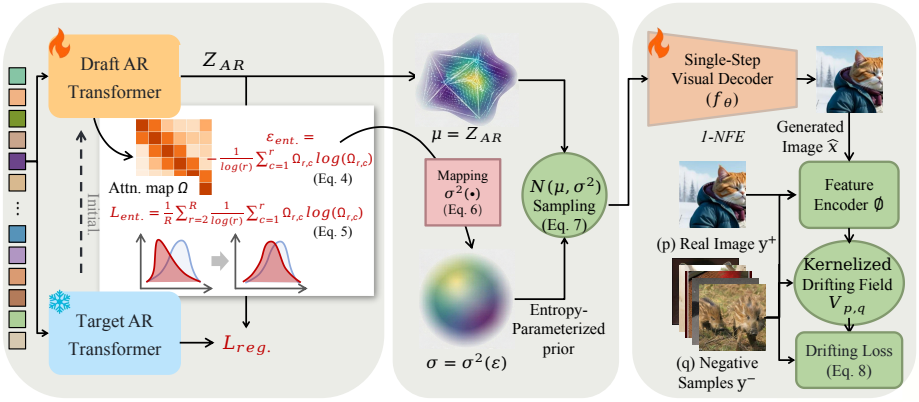

Fig. 3: Illustration of the proposed Drift-AR framework. (Left) Entropy-informed speculative decoding alleviates entropy mismatch between draft AR model and target AR model, which provides entropy-aligned semantic guidance that drives the draft AR model learns diverse, uncertainty-aware feature predictions rather than collapsing to overconfident modes. (Right) The visual decoder learns an anti-symmetric drifting field V_θ guided by Entropy-Parameterized Prior over the pushforward distribution q; the training procedure evolves q toward the data distribution p, thus when equilibrium the drift vanishes and *single-step* (1-NFE) generation is achieved. The overall Drift-AR framework is end-to-end training, which couples entropy-guided AR with drifting-field vision decoder optimization.

In contrast, *drift-based* generative models [5] avoid time-step-based iterative denoising. A network f_θ maps noise $\epsilon \sim p_\epsilon$ to samples $x = f_\theta(\epsilon)$; the pushforward distribution $q = (f_\theta)_{\#}p_\epsilon$ is evolved during training via a **drifting field** $V_{p,q}(x)$ so that q approaches the data distribution p. When V is **anti-symmetric** ($V_{p,q} = -V_{q,p}$), equilibrium $q = p$ implies $V = 0$, enabling **single-step (1-NFE)** generation at inference without iterative refinement. This avoids distillation-based acceleration (e.g., DMD [39]) and its need for a multi-step teacher.

4 Method

As shown in Fig. 3, Drift-AR is organized around a single guiding signal—the per-position prediction entropy $\mathcal{E}^{(r)}_{\text{entropy}}$—which drives both AR acceleration and single-step visual decoding. First, entropy informs speculative decoding by aligning draft–target entropy distributions (Sec. 4.1). The same entropy is then reinterpreted as the physical variance of an entropy-parameterized prior for an anti-symmetric drifting field, enabling 1-NFE generation (Sec. 4.2). A unified pipeline ties both components together (Sec. 4.3).

4.1 Entropy-Informed Speculative Decoding

Speculative decoding is a natural candidate for accelerating the AR stage of AR-Diffusion models, but direct application of EAGLE [15] yields poor performance. The draft model's outputs are frequently rejected by the target, and its low-entropy (overconfident) predictions lead to diminished diversity (Fig. 2(a)). The root cause is *entropy mismatch*: the small model produces overly confident features that deviate from the target's entropy distribution. As shown in the left of Fig. 3, we adapt EAGLE to the continuous latent space with two changes.

Continuous-space regression loss. We remove EAGLE's discrete classification loss $\mathcal{L}_{\text{cls}}$ (Eq. 2) and replace its token-conditioned regression with a continuous-feature variant. The draft is conditioned on the continuous AR feature sequence rather than discrete output tokens:

$$\mathcal{L}_{\text{reg}} = \text{Smooth L1}(z_{j+1},\ \text{DraftModel}(Z_{1:j},\ F_{1:j}))\,, \tag{3}$$

where $Z_{1:j}$ is the sequence of continuous AR features produced so far, $F_{1:j}$ is the target model's hidden-state sequence, and z_{j+1} is the target's next-step feature. This preserves structural alignment between draft and target while remaining consistent with continuous-space AR.

Causal-normalized entropy loss. In a causal-masked attention matrix, row r attends only to row $1, \ldots, r$, so its maximum possible entropy is $\log(r)$. A plain average over rows would unfairly penalize early rows (e.g., r=1 has entropy zero by construction) and distort gradients. We normalize each row's negative entropy by $\log(r)$:

$$\mathcal{E}^{(r)}_{\text{entropy}} \triangleq -\frac{1}{\log(r)} \sum_{c=1}^{r} \Omega_{r,c} \log \Omega_{r,c}, \tag{4}$$

where per-position entropy indicator $\mathcal{E}^{(r)}_{\text{entropy}} \in [0, 1]$, and total entropy loss is calculated as:

$$\mathcal{L}_{\text{entropy}} = -\frac{1}{R} \sum_{r=2}^{R} \mathcal{E}^{(r)}_{\text{entropy}}, \tag{5}$$

where $\Omega_{r,c}$ is the (r, c)-th softmax entry of the penultimate attention map ($\Omega_{r,c}$=0 for c>r; we start at r=2 since row r=1 has a single support and zero entropy). Minimizing $\mathcal{L}_{\text{entropy}}$ maximizes the *normalized* attention entropy at every position, so that training and the length-normalized inference threshold (Sec. 4.3) use the same scale.

Note that $\mathcal{E}^{(r)}_{\text{entropy}}$ has a dual role: it regularizes the draft to match the target's entropy during speculative decoding, and it is reused as the *physical variance* of the r-th AR feature when building the initial distribution for the drifting decoder (Sec. 4.2), tying AR uncertainty directly to the generative process.

4.2 Entropy-Parameterized Single-Step Generative Drifting

In typical AR-Diffusion hybrids (e.g., MAR, TransDiff, NextStep-1), the visual decoder is conditioned on AR features but its initial state is drawn from a fixed

Gaussian prior unrelated to the AR prediction's confidence; the visual decoder does not treat AR uncertainty as part of the generative state.

We instead treat the AR output as the *physical* initial distribution for a single-step drifting process: the prior q is defined directly from the AR features, with variance tied to per-position entropy, where entropy is related to the determinacy of generation—higher entropy indicates greater uncertainty like what we have shown in Fig. 2(c)(d), so that confident predictions yield low-variance initial states and uncertain ones yield higher variance.

Mapping $\mathcal{E}^{(r)}_{\text{entropy}}$ to vision decoder initial distribution. As shown in the middle of Fig. 3, we first map the normalized per-position entropy $\mathcal{E}^{(r)}_{\text{entropy}} \in [0,1]$ to a non-negative variance scale. Since entropy reflects structural uncertainty in the AR process, we use a monotonic mapping so that higher entropy corresponds to larger variance. Specifically, we adopt a bounded exponential temperature mapping:

$$\sigma(\mathcal{E}) = \sigma_{\max} \cdot \frac{e^{\mathcal{E}/\tau_\sigma} - 1}{e^{\mathcal{E}_{\max}/\tau_\sigma} - 1}, \tag{6}$$

where $\sigma_{\max}$ caps the variance, τ_σ controls curvature, and $\mathcal{E}_{\max}$ normalizes the input range to $[0, \sigma_{\max}]$ ($\sigma_{\max}$=0.5, τ_σ=2.0 in experiments). When $\mathcal{E}$ is small (confident AR), $\sigma(\mathcal{E})$ remains close to zero; as $\mathcal{E}$ approaches 1, the variance smoothly increases toward $\sigma_{\max}$.

The R=h×w AR features are reshaped into a 2D entropy map $\mathbf{E} \in \mathbb{R}^{h\times w}$ with $\mathbf{E}_{i,j} = \mathcal{E}^{(r)}_{\text{entropy}}$ (r=i·w+j). At each position r, we construct a Gaussian prior centered at the AR feature with entropy-determined variance:

$$q\Big(x^{(r)} \mid z^{(r)}_{AR}, \mathcal{E}^{(r)}_{\text{entropy}}\Big) = \mathcal{N}\Big(x^{(r)}; z^{(r)}_{AR}, \sigma^2\Big(\mathcal{E}^{(r)}_{\text{entropy}}\Big)\,\mathbf{I}\Big). \tag{7}$$

Thus, positions with low entropy (confident AR grounding) produce sharply concentrated initial states, whereas high-entropy positions yield broader initial distributions.

We sample $x_0^{(r)}$ from $q^{(r)}$ via the standard Gaussian reparameterization $x_0^{(r)} = z^{(r)}_{AR} + \sigma(\mathcal{E}^{(r)}_{\text{entropy}})\,\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$, yielding the initial decoder state $x_0 \in \mathbb{R}^{h\times w\times d}$.

Aligning AR latents with image feature via drifting field. As shown in the right of Fig. 3, we train a visual decoder f_θ with x_0 sampled from the prior q as input and c as the condition. The entropy map $\mathbf{E}$ is injected via additional Entropy-AdaLN. The corresponding output is $\hat{\mathbf{x}} = f_\theta(x_0, c, \mathbf{E})$, through which the decoder aligns the AR latents with the VAE-constrained image latent space.

To avoid dimension mismatch between the decoder's latent output and the drifting field defined in feature space, we map $\hat{\mathbf{x}}$ into the corresponding latent space using a frozen feature encoder ϕ. For simplicity and effectiveness, we use a copy of the target AR model's penultimate layer as ϕ, which is fixed throughout training. The loss is therefore formulated entirely in ϕ-space, following the kernelized attraction–repulsion construction of the drifting field $V_{p,q}$:

$$\mathcal{L}_{\text{drift}} = \mathbb{E}_{x\sim q}\left[\Big\|\phi(\hat{\mathbf{x}}) - \text{stopgrad}\Big(\phi(\hat{\mathbf{x}}) + V_{p,q}\big(\phi(\hat{\mathbf{x}})\big)\Big)\Big\|_2^2\right]. \tag{8}$$

Specifically, with prediction and regression target constrained in same feature space, $V_{p,q}$ is computed following drifting model [5]: for each generated sample, real samples exert an *attraction* with field V_p^+, and generated samples *repulsion* with V_q^-, where p, q represents real data and generated samples, respectively. With kernel functions at multiple temperatures for characterizing similarities among positive and negative samples, the net force yields the drift direction.

Gradients flow through $\phi(\hat{\mathbf{x}})$ and the decoder f_θ, while ϕ itself remains frozen. This design is physically self-consistent: high-entropy positions receive larger variance and therefore stronger drift, whereas low-entropy positions have near-zero variance where the field naturally vanishes. This property enables 1-NFE generation without iterative refinement.

4.3 Unified Training and Inference Pipeline

Annealed training for stabilizing the drifting field. A naive end-to-end training that jointly updates the AR module and the drifting visual decoder suffers from the "moving prior" problem: since the source distribution q is induced by the AR outputs, updating the AR module changes q at every iteration. As a result, the drifting field is effectively asked to transport samples toward a continuously shifting target, which makes the drifting dynamics unstable.

This is particularly problematic because the anti-symmetric drifting field $V_{p,q}$ is well-defined only when both the source q and the target p are stationary, in which case $q{=}p \Rightarrow V_{p,q}{=}0$ corresponds to a true equilibrium. When q keeps moving, this equilibrium interpretation breaks. We therefore adopt a two-phase schedule controlled by a scalar weight $\alpha(t)$:

$$\alpha(t) = \begin{cases} \alpha_0 - \alpha_0 \cdot \dfrac{t}{T_{\text{freeze}}}, & t \le T_{\text{freeze}}, \\ 0, & t > T_{\text{freeze}}, \end{cases} \tag{9}$$

where $\alpha_0 = 0.95$ and $T_{\text{freeze}} = 0.8 \cdot T_{\text{total}}$. The total loss is:

$$\mathcal{L}_{\text{total}} = \alpha(t) \cdot (\mathcal{L}_{reg} + \mathcal{L}_{entropy}) + (1 - \alpha(t)) \cdot \mathcal{L}_{\text{drift}}, \tag{10}$$

with $\mathcal{L}_{\text{drift}}$ (Eq. 8), $\mathcal{L}_{\text{reg}}$ (Eq. 3), and $\mathcal{L}_{\text{entropy}}$ (Eq. 5). Accordingly, the training process is divided into two phases:

Phase I ($t \le T_{freeze}$): Joint optimization. $\alpha(t)$ decays linearly from α_0 to 0. We initialize $\alpha_0{<}1$ (0.95 rather than 1) so that the drifting decoder receives a small supervisory signal from the beginning, avoiding cold-start collapse. During this phase, the AR prior is optimized with $\mathcal{L}_{\text{reg}}$ and $\mathcal{L}_{\text{entropy}}$, while the drifting decoder simultaneously learns to correct the evolving source distribution q toward the target distribution p. By the end of Phase I, the AR-induced prior q becomes sufficiently stable for subsequent drifting-field optimization.

Phase II ($t > T_{freeze}$): Pure drifting. $\alpha(t) = 0$, so only $\mathcal{L}_{\text{drift}}$ remains active. To keep the source distribution q stationary, the AR prior is frozen in Phase II when constructing x_0, preventing $\mathcal{L}_{\text{drift}}$ from back-propagating into the AR module and avoiding potential gradient leakage introduced by the reparameterization

(Sec. 4.2). The drifting decoder then optimizes $V_{p,q}$ over a fixed source distribution q, restoring the stationary-source assumption required by the equilibrium condition of drifting dynamics.

Context-aware early-stopping for efficient speculation. During speculative decoding we monitor the draft's attention entropy $\mathcal{E}^{(r)}_{\text{entropy}}$ to decide when to stop speculation and fall back to the target. We use the *shallow*-layer entropy [36], which correlates with prediction quality and is computationally efficient. When the normalized entropy falls below a threshold, speculation stops and the target model verifies the draft prefix in parallel [11]. The target accepts the longest matching prefix, discards the rest, and continues autoregressive generation from the accepted position. This avoids wasting target forward passes on low-quality draft segments. The threshold τ_{global} must handle two issues:

Length normalization. Based on the normalized entropy $\mathcal{E}^{(r)}_{\text{entropy}}$ at position r (denoted as $\mathcal{E}^r$) in Eq. 4, we set a global baseline $\tau_{\text{global}} = 0.3\,\mathbb{E}[\mathcal{E}^r] - 0.1\,\text{std}(\mathcal{E}^r)$ from training statistics, so that the draft acceptance policy is neither overly conservative nor aggressive, achieving a better balance between generation efficiency and quality.

Content-aware adaptation. A constant τ_{global} can trigger premature or late stopping in inherently low-entropy regions (e.g., smooth sky). To make the stopping criterion content-aware, we set the threshold based on the target model's normalized entropy. Specifically, at context length c we compute the target entropy $\hat{\mathcal{E}}^{\text{target}}_c$ and maintain an EMA $\bar{\mathcal{E}}^r_{\text{target}} = 0.9\,\bar{\mathcal{E}}^r_{\text{target}} + 0.1\,\mathcal{E}^r_{\text{target}}$ with initialization $\bar{\mathcal{E}}^{\text{target}}_0 = \tau_{\text{global}}/\gamma$. We then define a dynamic stopping threshold $\tau_c = \gamma \cdot \bar{\mathcal{E}}^{\text{target}}_c$ where γ=0.8. We stop speculation when the draft entropy $\mathcal{E}^r_{\text{entropy}} < \tau_c$. The threshold thus tightens in complex, high-entropy regions and relaxes in smooth ones, with no extra cost because the target already computes entropy during verification.

5 Experiments

5.1 Implementation Details

To validate the effectiveness of Drift-AR, we conduct extensive experiments on representative AR-Diffusion hybrids. We cover a broad range of architectures: MAR [13], TransDiff [42], and NextStep-1 [31]. Our method achieves **1-NFE** for the visual decoding stage: the drift-based decoder performs a single forward pass at inference, without timestep or iterative denoising.

Baselines. We compare against the original implementations and state-of-the-art acceleration methods. For AR acceleration: vanilla speculative decoding (SD) and LazyMAR [37] (MAR-specific). For diffusion acceleration: standalone DMD [39]. All DMD and LazyMAR results use their official configurations; SD uses 6-layer draft models initialized from the target, consistent across baselines.

Drift-AR training. We train the drifting field and AR components end-to-end in a single pipeline (no separate CD/DMD stages). For TransDiff: the draft AR has 6 transformer blocks; the drifting decoder replaces the original DiT with our

Table 1: Class-conditional generation on ImageNet 256×256. Drift-AR achieves the highest speedup while matching or improving FID and IS across all model scales.

Method	Latency/s	Speedup	FID ↓	IS ↑
MAR [13]				
MAR-L [13]	5.31	1.00	1.78	296.0
MAR-L [13] + DMD [39]	1.99	2.67	1.81	295.5
MAR-L [13] + LazyMAR [37]	2.29	2.32	1.93	297.4
MAR-L [13] + Ours	0.96	5.53	**1.76**	**297.4**
MAR-H [13]	9.97	1.00	1.55	303.7
MAR-H [13] + DMD [39]	4.17	2.39	1.73	301.0
MAR-H [13] + LazyMAR [37]	4.24	2.35	1.69	299.2
MAR-H [13] + Ours	1.93	5.16	**1.53**	**304.6**
TransDiff [42]				
TransDiff-L [42]	3.17	1.00	1.61	295.1
TransDiff-L [42] + SD [15]	1.75	1.81	1.88	283.6
TransDiff-L [42] + DMD [39]	1.61	1.97	1.79	288.3
TransDiff-L [42] + Ours	0.64	4.96	**1.61**	**295.8**
TransDiff-H [42]	6.72	1.00	1.55	297.9
TransDiff-H [42] + SD [15]	3.52	1.91	1.68	291.1
TransDiff-H [42] + DMD [39]	3.48	1.93	1.71	289.9
TransDiff-H [42] + Ours	1.33	5.06	**1.57**	**298.1**

drift-based generator. Phase I spans 80% of total epochs ($T_{\text{freeze}} = 0.8 \cdot T_{\text{total}}$) with α linearly decaying from 0.95 to 0; Phase II freezes the AR and optimizes only $\mathcal{L}_{\text{drift}}$. Key hyperparameters: $\sigma_{\max} = 0.5$, $\tau_\sigma = 2.0$, $\mathcal{E}_{\max}$ estimated from training entropy; kernel temperatures $\tau \in \{0.02, 0.05, 0.2\}$. Optimizer: AdamW ($\text{lr} = 1 \times 10^{-4}$, batch size 32 for ImageNet). All models are evaluated following the original papers. ImageNet: FID [9], IS [23], latency on 50,000 images. Text-to-image: GenEval [8], FID and CLIP on MJHQ-30K [12].

5.2 Comparison

ImageNet 256×256 (Class-Conditional). As shown in Table 1, Drift-AR consistently delivers the highest speedup while maintaining or improving generation quality. For **MAR**, our method attains 5.53× and 5.16× speedup on MAR-L and MAR-H respectively. On MAR-L, we improve both FID (1.76 vs. baseline 1.78) and IS (297.4 vs. 296.0); on MAR-H, we improve FID (1.53 vs. 1.55) and IS (304.6 vs. 303.7), with both metrics surpassing DMD and LazyMAR. DMD and LazyMAR achieve only ∼2.4× speedup with degraded or marginal quality. For **TransDiff**, Drift-AR reaches 4.96× and 5.06× speedup on the L and H variants. We improve IS (295.8, 298.1) relative to the baseline (295.1, 297.9). Vanilla SD suffers severe quality collapse (IS drops to 283.6 and 291.1); DMD improves speed but degrades FID/IS. The key advantage is **1-NFE** visual decoding: our drift-based generator dispenses with iterative denoising entirely, whereas DMD still requires multi-step or distilled sampling.

Text-to-Image (NextStep-1). On GenEval and MJHQ-30K (Table 2), Drift-AR achieves 3.81× speedup with GenEval 0.66, FID 6.66, and CLIP 29.02—all superior to the baseline (0.63, 6.71, 28.67). SD reaches 2.03× but loses quality (FID 6.90, CLIP 27.96); DMD collapses (FID 8.33, CLIP 25.19). Qualitative

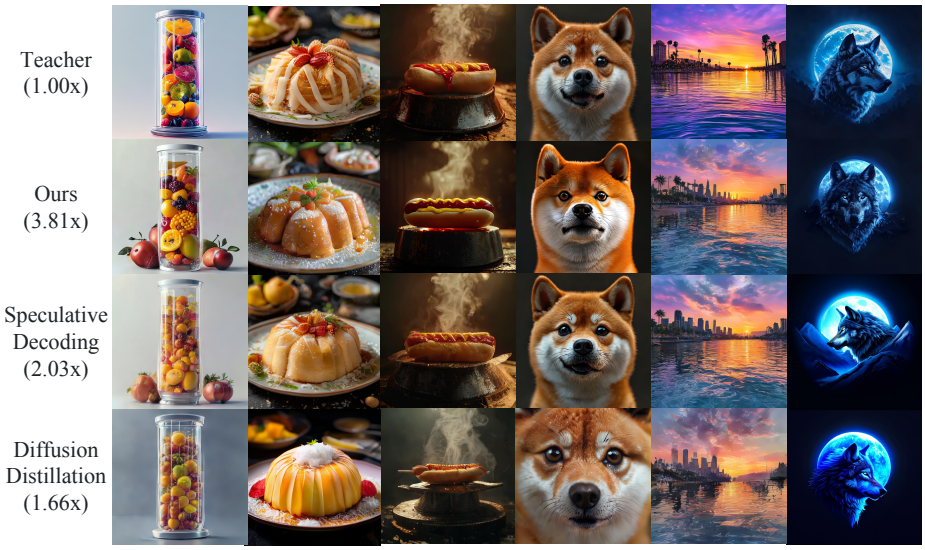

Fig. 4: Visual comparisons with NextStep-1 [31] on MJHQ-30K [12].

comparisons in Fig. 4 confirm that our method preserves fine-grained details and semantic consistency while dramatically reducing latency.

5.3 Ablation Study

We ablate five key components on TransDiff-H, with all other settings consistent with our full framework. Refer to the supplementary material for more details.
$\mathbb{A}$**: Ablating Entropy Parameterization.** We replace the entropy-parameterized variance $\sigma^2(\mathcal{E}_{entropy})$ with a constant variance for the initial drifting state, testing whether entropy as the physical variance is essential for the drifting field's equilibrium.
$\mathbb{B}$**: Ablating Stage-Wise Loss Reweighting.** We fix $\alpha = 0.5$ throughout training instead of the dynamic annealing $\alpha(t)$, examining whether adaptive reweighting is necessary to coordinate AR and drifting-field components.
$\mathbb{C}$**: Ablating the Anti-Symmetric Kernel Objective.** We keep only the *attraction* term and remove the *repulsion* term (non-antisymmetric variant). The resulting field no longer satisfies $q{=}p \Rightarrow V{=}0$, so drift need not vanish at equilibrium.
$\mathbb{D}$**: Ablating Entropy-Based Early Stopping in Inference.** We disable entropy monitoring during speculative decoding, assessing whether the early stopping mechanism mitigates residual low-quality draft output.
$\mathbb{E}$**: Ablating Prior Freezing in Phase II.** We allow $\mathcal{L}_{\text{drift}}$ gradients to backpropagate through the AR module in Phase II (i.e., we do not detach z_{AR} and **E**), testing the "moving prior" hypothesis.

Table 2: (left) Text-to-image comparison on GenEval and MJHQ-30K. (right) Ablation study on TransDiff-H (ImageNet 256×256): removing entropy parameterization (𝔸) causes the largest degradation, validating entropy as the core design.

Method	Speedup	GenEval ↑	FID ↓	CLIP ↑
NextStep-1 [31]	1.00	0.63	6.71	28.67
NextStep-1 [31] + SD [15]	2.03	0.63	6.90	27.96
NextStep-1 [31] + DMD [39]	1.66	0.59	8.33	25.19
NextStep-1 [31] + Ours	3.81	**0.66**	**6.66**	**29.02**

Method	FID ↓	IS ↑
TransDiff-H	1.55	297.9
Ours w/o 𝔸	1.72	289.3
Ours w/o 𝔹	1.69	292.6
Ours w/o ℂ	1.69	291.8
Ours w/o 𝔻	1.62	295.5
Ours w/o 𝔼	1.67	291.9
Ours (full)	**1.57**	**298.1**

Analysis of Ablation Results (Table 2). Removing any component degrades both FID and IS. *w/o* 𝔸 (entropy parameterization) produces the largest degradation: FID rises to 1.72 and IS drops to 289.3. This is the most critical ablation, as it directly validates the core thesis that entropy should parameterize the drifting prior: without entropy-driven variance, the drifting field receives a fixed-variance prior that cannot adapt to local generation difficulty (cf. Fig. 2(c)(d)). *w/o* 𝔹 (stage-wise reweighting): FID 1.69, IS 292.6. Fixing α=0.5 prevents the AR prior from stabilizing before Phase II; the drifting field is trained against a moving prior, undermining the equilibrium condition. *w/o* ℂ (anti-symmetric kernel): FID 1.69, IS 291.8. The non-antisymmetric variant lacks the guarantee $q{=}p \Rightarrow V{=}0$, leading to residual drift at convergence and lower-quality 1-NFE samples. *w/o* 𝔻 (early stopping): FID 1.62, IS 295.5. Disabling context-aware early stopping allows low-quality or overconfident draft outputs to propagate, increasing rejection overhead and slightly degrading diversity. *w/o* 𝔼 (prior freezing): FID 1.67, IS 291.9. Allowing drift gradients to update the AR in Phase II creates a moving prior that destabilizes the drifting field; the equilibrium condition $q{=}p \Rightarrow V{=}0$ is never satisfied stably, leading to noticeable quality degradation. The full model (1.57 FID, 298.1 IS) confirms that all five components contribute to the final performance.

5.4 Analysis

Decoder Step Analysis. A natural question is whether the drifting decoder truly needs only one forward pass. Table 3 compares three decoder types on TransDiff-H at varying step counts. TransDiff uses 20 denoising steps by default. The original decoder degrades rapidly when fewer steps are used, collapsing to FID 14.72 at 1 step. DMD recovers quality through distillation but still yields FID 2.93 at 1 step. In contrast, our drifting decoder achieves FID 1.57 at 1 step—on par with the 20-step diffusion baseline—confirming that the anti-symmetric drifting field concentrates the generative capacity into a single pass.

Sensitivity to $\sigma_{\max}$. The maximum variance $\sigma_{\max}$ in Eq. 6 controls the range of entropy-to-variance mapping. Table 4 shows that performance is robust across a moderate range (0.3–0.7), peaking at $\sigma_{\max}$=0.5. Too small a value (0.1) collapses the prior toward a Dirac delta, depriving the drifting field of sufficient

Table 3: Decoder step analysis on TransDiff-H (ImageNet 256×256). Our drifting decoder at 1 step matches the 20-step diffusion baseline, while DMD still degrades significantly at 1 step.

Decoder	Steps	Latency/s	Speedup	FID ↓	IS ↑
Diffusion	20 (default)	6.72	1.00×	1.55	297.9
Diffusion	8	2.85	2.36×	2.34	279.8
Diffusion	4	1.65	4.07×	3.89	261.5
Diffusion	1	1.28	5.25×	14.72	148.3
DMD [39]	4	1.85	3.63×	2.11	282.9
DMD [39]	1	1.30	5.17×	2.93	273.2
Drifting (Ours)	1	1.33	5.06×	**1.57**	**298.1**

Table 4: Sensitivity to $\sigma_{\max}$ on TransDiff-H (ImageNet 256×256). Performance is robust across 0.3–0.7; extremes hurt by under- or over-spreading the entropy-parameterized prior.

$\sigma_{\max}$	0.1	0.3	**0.5**	0.7	1.0
FID ↓	1.68	1.58	**1.57**	1.57	1.64
IS ↑	292.7	296.5	**298.1**	296.3	293.1

randomness to cover the data manifold. Too large a value (1.0) produces an overly diffuse prior that demands excessive drift, degrading quality.

Entropy–Prediction Error Correlation. Fig. 2(c)(d) quantifies the relationship between per-position AR entropy $\mathcal{E}^{(r)}$ and prediction error $\|z_{AR}^{(r)} - x_{gt}^{(r)}\|$ on the ImageNet validation set. The scatter plot shows a clear positive correlation (Pearson r=0.64), and the binned analysis confirms that mean error increases monotonically with entropy. This validates our entropy-parameterized variance: positions where the AR model is uncertain are precisely those where the decoder must apply the strongest correction, and our mapping (Eq. 6) allocates prior spread accordingly.

6 Discussion and Conclusion

We show that the per-position prediction entropy of continuous-space AR models, a signal previously overlooked, simultaneously governs draft quality in speculative decoding and encodes local generation difficulty for visual decoding. This dual role makes entropy the natural unifying principle for accelerating AR-Diffusion hybrids. Drift-AR exploits this by (1) aligning draft–target entropy distributions for efficient speculation, and (2) reinterpreting entropy as the physical variance of an anti-symmetric drifting field, achieving 1-NFE visual decoding without distillation. Experiments on MAR, TransDiff, and NextStep-1 demonstrate 3.8–5.5× speedup while matching or surpassing original quality.

Limitations. The entropy estimation relies on attention-map statistics, whose quality may degrade under distribution shift or for architectures without causal

attention. The two-phase annealed training requires a longer schedule than standard distillation to stabilize the prior before drifting-field optimization. Extending the framework to video or higher-resolution generation, where spatial entropy patterns become more complex, remains future work.

Acknowledgements

This work was supported by the Anhui Provincial Natural Science Foundation under Grant 2108085UD12. We acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC. The AI-driven experiments, simulations and model training were performed on the robotic AI-Scientist platform of Chinese Academy of Sciences.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: GPT-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Anil, R., Dai, A.M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al.: PaLM 2 technical report. arXiv preprint arXiv:2305.10403 (2023)
3. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your ViT but faster. arXiv preprint arXiv:2210.09461 (2022)
4. Bolya, D., Hoffman, J.: Token merging for fast stable diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4599–4603 (2023)
5. Deng, M., Li, H., Li, T., Du, Y., He, K.: Generative modeling via drifting. arXiv preprint arXiv:2602.04770 (2026)
6. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12873–12883 (2021)
7. Fang, G., Ma, X., Wang, X.: Structural pruning for diffusion models. arXiv preprint arXiv:2305.10924 (2023)
8. Ghosh, D., Hajishirzi, H., Schmidt, L.: Geneval: An object-focused framework for evaluating text-to-image alignment. Advances in Neural Information Processing Systems **36**, 52132–52152 (2023)
9. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local Nash equilibrium. Advances in Neural Information Processing Systems **30**, 6626–6637 (2017)
10. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems **33**, 6840–6851 (2020)
11. Leviathan, Y., Kalman, M., Matias, Y.: Fast inference from transformers via speculative decoding. arXiv preprint arXiv:2211.17192 (2023)
12. Li, D., Kamko, A., Akhgari, E., Sabet, A., Xu, L., Doshi, S.: Playground v2.5: Three insights towards enhancing aesthetic quality in text-to-image generation. arXiv preprint arXiv:2402.17245 (2024)

13. Li, T., Tian, Y., Li, H., Deng, M., He, K.: Autoregressive image generation without vector quantization. Advances in Neural Information Processing Systems **37**, 56424–56445 (2024)
14. Li, Y., Wei, F., Zhang, C., Zhang, H.: EAGLE-2: Faster inference of language models with dynamic draft trees. arXiv preprint arXiv:2406.16858 (2024)
15. Li, Y., Wei, F., Zhang, C., Zhang, H.: EAGLE: Speculative sampling requires rethinking feature uncertainty. arXiv preprint arXiv:2401.15077 (2024)
16. Li, Y., Wei, F., Zhang, C., Zhang, H.: EAGLE-3: Scaling up inference acceleration of large language models via training-time test. arXiv preprint arXiv:2503.01840 (2025)
17. Lin, S., Wang, A., Yang, X.: SDXL-Lightning: Progressive adversarial diffusion distillation. arXiv preprint arXiv:2402.13929 (2024)
18. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)
19. Liu, D., Zhao, S., Zhuo, L., Lin, W., Qiao, Y., Li, H., Gao, P.: Lumina-mGPT: Illuminate flexible photorealistic text-to-image generation with multimodal generative pretraining. arXiv preprint arXiv:2408.02657 (2024)
20. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: International Conference on Learning Representations. pp. 1–33 (2023)
21. Lou, J., Luo, W., Liu, Y., Li, B., Ding, X., Hu, W., Cao, J., Li, Y., Ma, C.: Token caching for diffusion transformer acceleration. arXiv preprint arXiv:2409.18523 (2024)
22. Ren, S., Yu, Q., He, J., Shen, X., Yuille, A., Chen, L.C.: FlowAR: Scale-wise autoregressive image generation meets flow matching. arXiv preprint arXiv:2412.15205 (2024)
23. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. Advances in Neural Information Processing Systems **29**, 2234–2242 (2016)
24. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv:2202.00512 (2022)
25. Shang, Y., Yuan, Z., Xie, B., Wu, B., Yan, Y.: Post-training quantization on diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1972–1981 (2023)
26. So, J., Lee, J., Ahn, D., Kim, H., Park, E.: Temporal dynamic quantization for diffusion models. Advances in Neural Information Processing Systems **36**, 48686–48698 (2023)
27. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International Conference on Machine Learning. pp. 2256–2264. PMLR (2015)
28. Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency models. In: International Conference on Machine Learning. pp. 32211–32252. PMLR (2023)
29. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
30. Sun, P., Jiang, Y., Chen, S., Zhang, S., Peng, B., Luo, P., Yuan, Z.: Autoregressive model beats diffusion: Llama for scalable image generation. arXiv preprint arXiv:2406.06525 (2024)
31. Team, N., Han, C., Li, G., Wu, J., Sun, Q., Cai, Y., Peng, Y., Ge, Z., Zhou, D., Tang, H., et al.: NextStep-1: Toward autoregressive image generation with continuous tokens at scale. arXiv preprint arXiv:2508.10711 (2025)

32. Tian, K., Jiang, Y., Yuan, Z., Peng, B., Wang, L.: Visual autoregressive modeling: Scalable image generation via next-scale prediction. Advances in Neural Information Processing Systems **37**, 84839–84865 (2024)
33. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: LLaMA: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
34. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. Advances in Neural Information Processing Systems **30**, 6306–6315 (2017)
35. Wang, Y., Ren, S., Lin, Z., Han, Y., Guo, H., Yang, Z., Zou, D., Feng, J., Liu, X.: Parallelized autoregressive visual generation. arXiv preprint arXiv:2412.15119 (2024)
36. Wang, Z., Zhang, R., Ding, K., Yang, Q., Li, F., Xiang, S.: Continuous speculative decoding for autoregressive image generation. arXiv preprint arXiv:2411.11925 (2024)
37. Yan, F., Wei, Q., Tang, J., Li, J., Wang, Y., Hu, X., Li, H., Zhang, L.: LazyMAR: Accelerating masked autoregressive models via feature caching. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15552–15561 (2025)
38. Yin, T., Gharbi, M., Park, T., Zhang, R., Shechtman, E., Durand, F., Freeman, B.: Improved distribution matching distillation for fast image synthesis. Advances in Neural Information Processing Systems **37**, 47455–47487 (2024)
39. Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W.T., Park, T.: One-step diffusion with distribution matching distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6613–6623 (2024)
40. Yuan, Z., Zhang, H., Lu, P., Ning, X., Zhang, L., Zhao, T., Yan, S., Dai, G., Wang, Y.: DiTFastAttn: Attention compression for diffusion transformer models. arXiv preprint arXiv:2406.08552 (2024)
41. Zhang, D., Li, S., Chen, C., Xie, Q., Lu, H.: Laptop-Diff: Layer pruning and normalized distillation for compressing diffusion models. arXiv preprint arXiv:2404.11098 (2024)
42. Zhen, D., Qiao, Q., Zheng, X., Yu, T., Wu, K., Zhang, Z., Liu, S., Yin, S., Tao, M.: Marrying autoregressive transformer and diffusion with multi-reference autoregression. arXiv preprint arXiv:2506.09482 (2025)