Beyond Next-Token: Next-X Prediction for Autoregressive Visual Generation

Sucheng Ren[1] Qihang Yu[2] Ju He[2] Xiaohui Shen[2] Alan Yuille[1] Liang-Chieh Chen[2]
[1]Johns Hopkins University [2]ByteDance

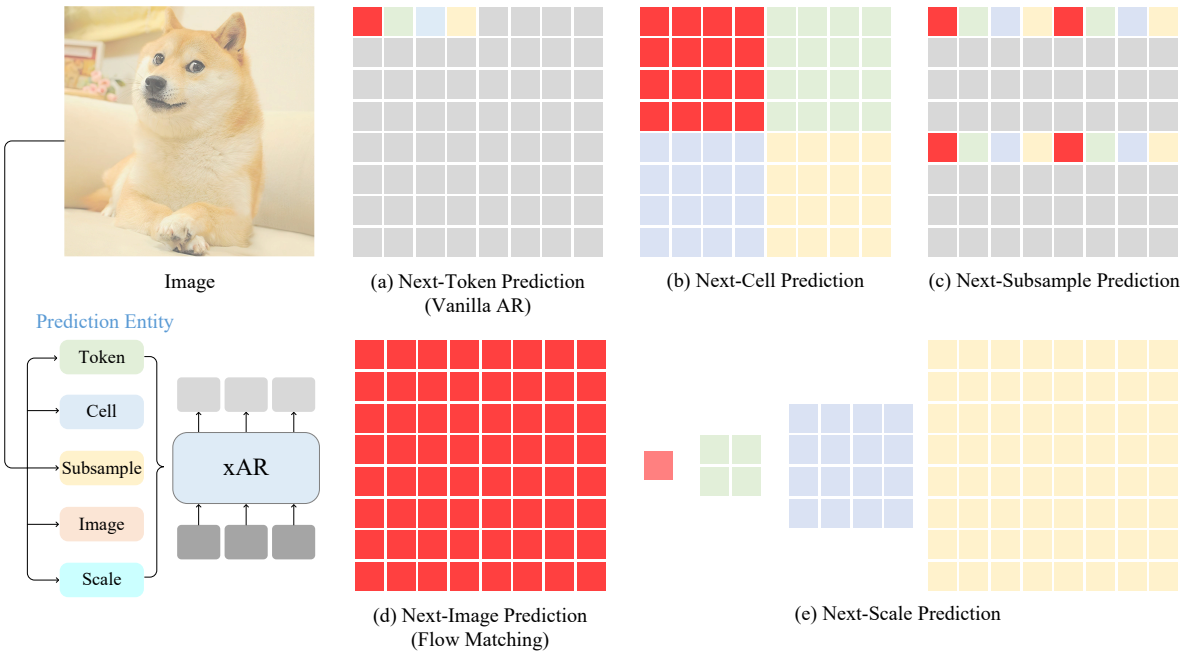

Figure 1. **xAR: Autoregressive (AR) Visual Generation with Next-X Prediction.** The proposed xAR adopts a general next-X prediction framework, where X is a flexible prediction entity that can correspond to: (a) an individual image patch (as in vanilla AR [11]), (b) a cell (a group of spatially contiguous tokens), (c) a subsample (a non-local grouping), (d) an entire image (as in flow-matching [28]), or (e) a scale (coarse-to-fine resolution, similar to VAR [55]). We use red, green, blue, yellow to illustrate the first four AR prediction steps for each entity example. The gray tokens represent the remaining tokens.

Abstract

*Autoregressive (AR) modeling, known for its next-token prediction paradigm, underpins state-of-the-art language and visual generative models. Traditionally, a "token" is treated as the smallest prediction unit, often a discrete symbol in language or a quantized patch in vision. However, the optimal token definition for 2D image structures remains an open question. Moreover, AR models suffer from exposure bias, where teacher forcing during training leads to error accumulation at inference. In this paper, we propose xAR, a generalized AR framework that extends the notion of a token to an entity X, which can represent an individual patch token, a cell (a $k \times k$ grouping of neighboring patches), a subsample (a non-local grouping of distant patches), a scale (coarse-to-fine resolution), or even a whole image. Additionally, we reformulate discrete token classification as **continuous entity regression**, leveraging flow-matching methods at each AR step. This approach conditions training on noisy entities instead of ground truth tokens, leading to Noisy Context Learning, which effectively alleviates exposure bias. As a result, xAR offers two key advantages: (1) it enables flexible prediction units that capture different contextual granularity and spatial structures, and (2) it mitigates exposure bias by avoiding reliance on teacher forcing. On ImageNet-256 generation benchmark, our base model, xAR-B, outperforms DiT-XL/SiT-XL while achieving 20× faster inference. Meanwhile, xAR-H sets a new state-of-the-art with an FID of 1.24, running 2.2× faster than the previous best-performing model without relying on vision foundation modules (e.g., DINOv2) or advanced guidance interval sampling. Codes is publicly available at https://oliverrensu.github.io/project/xAR.*

1. Introduction

Autoregressive (AR) models have driven major advances in natural language processing (NLP) through next-token prediction, where each token is generated from its preceding tokens. This framework enables coherent, context-aware text generation, with landmark models like GPT-3 [5] and its successors [33, 34] setting new benchmarks across diverse NLP applications.

Building on the successes of AR modeling in NLP, researchers have extended this framework to computer vision, particularly for high-fidelity image generation [11, 23, 27, 54, 66, 69]. In these approaches, image patches are discretized into tokens [59] and reshaped into 1D sequences, allowing AR models to predict each token sequentially. However, unlike language, where tokens correspond to semantically meaningful units such as words, vision lacks a universally agreed-upon token definition. This naturally raises the question: *How can "next-token prediction" be generalized to "next-X prediction," and what constitutes the most suitable X for image generation?*

Additionally, beyond token design, traditional AR models rely on teacher forcing [63] during training, where ground truth tokens are provided at each step instead of the model's own predictions. While this stabilizes training, it introduces exposure bias [38], since the model is never exposed to potential errors. Consequently, during inference, without ground truth guidance, errors accumulate over time, leading to cascading errors and context drift as the model conditions solely on its past predictions.

To address these challenges, we propose xAR, a general next-X prediction framework that reformulates discrete token *classification* (conditioned on all preceding discrete *ground truth* tokens) into a continuous entity *regression* problem conditioned on all previous *noisy* entities. The regression process is guided by flow-matching [28, 30] at each AR step. As illustrated in Fig. 1, within this framework, we explored different prediction entity X, including an individual patch token, a cell (a group of surrounding tokens), a subsample (a non-local grouping), a scale (coarse-to-fine resolution), or even an entire image.

Unlike teacher forcing [63], which always provides ground truth inputs, xAR deliberately exposes the model to noisy contexts during training, allowing it to learn from imperfect, corrupted, or partially inaccurate conditions. We refer to this approach as Noisy Context Learning (NCL), a reformulation that reduces reliance on ground truth inputs, improving robustness and mitigating exposure bias [38] by enabling the model to generalize better during inference.

We demonstrate the effectiveness of xAR on the challenging ImageNet generation benchmark [9]. Through systematic experimentation with different X configurations, we find that ***next-cell*** prediction—where neighboring tokens are grouped into moderately sized cells (*e.g.*, 8×8 tokens)—yields the best performance by capturing richer spatial-semantic relationships. Leveraging both next-cell prediction and Noisy Context Learning, our base model xAR-B (172M) outperforms the large DiT-XL [36] and SiT-XL [32] (675M) while achieving 20× faster inference. Additionally, our largest model, xAR-H (1.1B), sets a new state-of-the-art with an FID of 1.24 and runs 2.2× faster than the previous best-performing model [70] on ImageNet-256 [9], without relying on vision foundation models (*e.g.*, DINOv2 [35]) or extra guidance interval sampling [25].

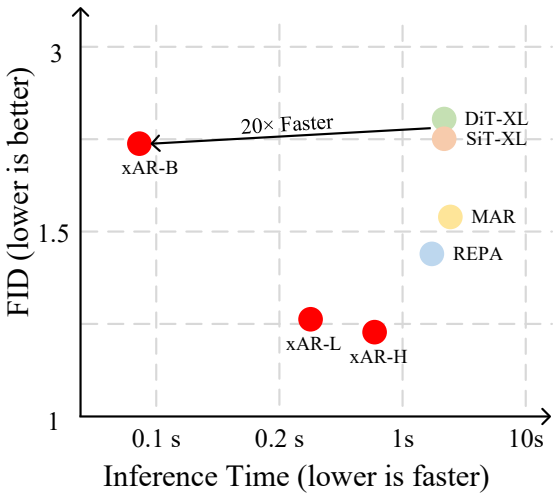

Figure 2. **ImageNet-256 Results.** Our base model, xAR-B, outperforms DiT-XL [36] and SiT-XL [32] while achieving 20× faster inference, and our largest model, xAR-H, establishes a new state-of-the-art with an FID of 1.24 on ImageNet-256.

2. Related Work

AR Modeling in NLP. Autoregressive language models [5, 33, 34, 37, 56] have driven significant progress toward general-purpose AI. Their core principle is simple yet powerful: predicting the next token based on preceding context. This approach has demonstrated impressive scalability, guided by scaling laws, and adaptability, enabling zero-shot generalization. These strengths have extended AR modeling beyond traditional language tasks, influencing a wide range of modalities.

AR Modeling in Vision. Inspired by the success of AR modeling in NLP, researchers have explored its application in vision [7, 39, 42–44, 46, 54, 55, 57, 66, 69]. A pioneering effort in this direction was PixelCNN [57], which factorized the joint pixel distribution into a product of conditionals, enabling the model to learn complex image distributions. This idea was further refined in PixelRNN [58], which incorporated recurrent layers to capture richer context in both horizontal and vertical directions. iGPT [7] extended this pixel-level approach by leveraging Transformers [60] for next-pixel prediction. Beyond next-pixel mod-

eling, AR methods have shifted toward more abstract token representations. VQ-VAE [59] introduced discrete latent codes that could be modeled autoregressively, offering a compressed yet expressive representation of images. Later models like Parti [66] and LlamaGen [54] combined these learned tokens with Transformer-based architectures to generate high-fidelity images while maintaining scalable training. Recently, MAR [27] introduced a diffusion-based approach [19, 53] to model per-token probability distributions in a continuous space, replacing categorical cross-entropy with a diffusion loss. VAR [55] extended next-token prediction to a coarse-to-fine scale prediction paradigm, progressively refining image details. Our work unifies these approaches under a general next-X prediction framework, where X can flexibly represent tokens, scales, or our newly introduced cells, providing a more flexible and generalizable formulation for autoregressive visual modeling.

Diffusion and Flow Matching. Beyond autoregressive modeling, diffusion [19, 45, 52, 53] and flow matching [12, 16, 28, 30, 40, 41, 51, 64] have surpassed Generative Adversarial Networks (GANs) [14, 50] by employing multi-step denoising. Latent Diffusion Models (LDMs) [47] improve speed and scalability by operating in a compressed latent space [24] instead of raw pixels. Building on this, DiT [36] and U-ViT [3] replace the traditional convolution-based U-Net [48] with Transformers [60] in latent space, further enhancing performance. Simple Diffusion [20, 21] introduces a streamlined approach for scaling pixel-space diffusion models to high-resolution outputs, while DiMR [29] progressively refines features across multiple scales, improving detail from low to high resolution. In parallel, flow matching [28, 30] reformulates the generative process by directly mapping data distributions to a standard normal distribution, simplifying the transition from noise to structured data. SiT [32] builds on this by integrating flow matching into DiT's Transformer backbone for more efficient distribution alignment. Extending this approach, SD3 [12] introduces a Transformer-based architecture that leverages flow matching for text-to-image generation. REPA [70] refines denoising by aligning noisy intermediate states with clean image embeddings extracted from pretrained visual encoders [35].

3. Method

In this section, we first provide an overview of autoregressive modeling with the next-token prediction paradigm in Sec. 3.1, followed by our proposed xAR framework with next-X prediction and Noisy Context Learning in Sec. 3.2.

3.1. Preliminary: Next-Token Prediction

Autoregressive modeling with next-token prediction is a fundamental approach in language modeling where the joint probability of a token sequence is factorized into a product of conditional probabilities. Formally, given a sequence $\boldsymbol{x} = \{x_1, x_2, \ldots, x_N\}$, the model estimates

$$P(\boldsymbol{x}) = \prod_{n=1}^{N} P\big(x_n \mid x_1, x_2, \ldots, x_{n-1}\big). \tag{1}$$

In practice, an autoregressive language model predicts the next token x_n through token classification, conditioned on all preceding tokens $\{x_1, x_2, \ldots, x_{n-1}\}$. This process proceeds sequentially from left to right (*i.e.*, $n = \{1, \ldots, N\}$) until the full sequence is generated. For visual generation, a VQ tokenizer [11, 59] discretizes an image into a sequence of tokens. An autoregressive visual generation model then follows the next-token prediction paradigm, sequentially predicting tokens through classification conditioned on previously generated tokens. However, directly applying the next-token prediction paradigm to visual generation introduces several challenges:

Information Density. In NLP, each token (*e.g.*, a word) carries rich semantic meaning. In contrast, visual tokens typically represent small image patches, which may not be as semantically meaningful in isolation. A single patch can contain fragments of different objects or textures, making it difficult for the model to infer meaningful relationships between consecutive patches. Additionally, the quantization process in VQ-VAE [59] can discard fine details, leading to lower-quality reconstructions. As a result, even if the model predicts the next token correctly, the generated image may still appear blurry or lack detail.

Accumulated Errors. Teacher forcing [63], a common training strategy, feeds the model ground truth tokens to stabilize learning. However, this reliance on perfect context causes exposure bias [2, 17]—the model never learns to recover from its potential mistakes. During inference, when it must condition on its own predictions, small errors can accumulate over time, leading to compounding artifacts and degraded output quality.

To address these challenges, we extend next-token prediction to *next-X prediction*, transitioning from traditional AR to xAR. This is accomplished by introducing a more expressive prediction entity X and training the model with noisy entities for improved robustness.

3.2. The Proposed xAR

We introduce xAR, which consists of two key components: next-X prediction (Sec. 3.2.1) and Noisy Context Learning (Sec. 3.2.2). We first detail each component, then describe the inference strategy (Sec. 3.2.3), followed by a discussion on how xAR enhances visual generation (Sec. 3.2.4).

3.2.1. Next-X Prediction

Given an image, we use an off-the-shelf VAE [24] (instead of VQ-VAE [59] to avoid quantization loss) to convert it into a continuous latent $I \in \mathcal{R}^{\frac{H}{f} \times \frac{W}{f} \times C}$, where H and W

denote image height and width, f is the downsampling rate (we use $f = 16$ [27]), and C represents the number of channels. We then construct a sequence of prediction entities $\boldsymbol{X} = \{X_1, X_2, \ldots, X_N\}$ based on I. Each X_i is a flexible entity that can represent an individual token (an image patch), a cell (a group of surrounding tokens), a subsample (a non-local grouping), a scale (coarse-to-fine resolution), or even an entire image. We outline common choices for X below and refer readers to Fig. 1 for visualization and Algorithm 1 for a PyTorch pseudo-code implementation.

Individual Patch Token (Fig. 1 (a)). When X_i corresponds to a single image patch, xAR reduces to standard AR modeling, where each token is predicted sequentially.

Cell (Fig. 1 (b)). The image is divided into an $m \times m$ grid, where each cell has $k \times k$ spatially adjacent tokens[1].

Subsample (Fig. 1 (c)). Entities are created by spatially and uniformly subsampling the image grid [11].

Entire Image (Fig. 1 (d)). As an extreme case, all tokens are grouped into a single entity, *i.e.*, $X = X_1 = I$, transforming xAR into a flow matching method [28, 30].

Scale (Fig. 1 (e)). A multi-scale hierarchical representation is constructed, similar to VAR [55]. Given any scale design $\{s_1, \ldots, s_N\}$, we define $X_i = \text{resize}(I, s_i)$, where resize refers to resizing the latent I to the target scale s_i. By default, we set $X_N = I \in \mathcal{R}^{\frac{H}{f} \times \frac{W}{f} \times C}$ (*i.e.*, $s_N = \frac{H}{f}$), and define $X_i = \text{resize}(I, \frac{H}{f} \cdot \frac{1}{2^{N-i}})$ (*i.e.*, $s_i = \frac{H}{f} \cdot \frac{1}{2^{N-i}}$) which progressively refines predictions from coarse to fine scales. Unlike VAR [55], our approach generalizes next-scale prediction to any scale configuration and does not require a specially designed multi-scale VQGAN tokenizer.

Default Choice of X. Extensive ablation studies in Sec. 4.2 show that cell (with a size of 8×8 tokens) achieves the best performance among all X designs. Therefore, unless specified otherwise, xAR adopts 8×8 cells as the default X.

3.2.2. Noisy Context Learning

xAR transitions the paradigm from "discrete token classification" (conditioned on all preceding *ground truth* tokens) to "continuous entity regression" (conditioned on all previous *noisy* entities). Specifically, unlike traditional AR modeling, which directly classifies X_n based on all preceding ground truth entities $\{X_1, \ldots, X_{n-1}\}$, xAR predicts X_n by minimizing a regression loss derived from flow matching [28, 30], conditioned on all previous noisy entities.

During training, we randomly sample n noise time steps $\{t_1, \ldots, t_n\} \subset [0, 1]$, and draw n noise samples $\{\epsilon_1, \ldots, \epsilon_n\}$ from the source Gaussian noise distribution. Specifically, at the n-th AR step, the noise samples are drawn as $\epsilon_n \sim \mathcal{N}(0, I)$, where ϵ_n and X_n share the same shape [47]. We construct the interpolated input $F_n^{t_n}$ as:

$$F_n^{t_n} = (1 - t_n)X_n + t_n\epsilon_n. \tag{2}$$

Note that in F, the superscript denotes the flow-matching noise time step, while the subscript represents the AR time step. We then define the velocity $V_n^{t_n}$ as:

$$\begin{aligned} V_n^{t_n} &= \frac{dF_n^{t_n}}{dt_n} \\ &= \epsilon_n - X_n, \end{aligned} \tag{3}$$

where $V_n^{t_n}$ represents the directional flow from $F_n^{t_n}$ toward X_n, guiding the transformation from the source to the target distribution.

The model is trained to predict the velocity $V_n^{t_n}$ using all preceding and current noisy entities $\{F_1^{t_1}, \ldots, F_n^{t_n}\}$:

$$\mathcal{L} = \sum_{n=1}^{N} \left\| \text{xAR}\left(\{F_1^{t_1}, \ldots, F_n^{t_n}\}, t_n; \theta\right) - V_n^{t_n} \right\|^2, \tag{4}$$

where xAR denotes our xAR model parameterized by θ.

We refer to this scheme as Noisy Context Learning (NCL), where the model is trained by conditioning on all previous noisy entities rather than perfect ground truth inputs. This effectively reduces reliance on clean training signals, improving robustness and mitigating exposure bias [38]. Fig. 3 (Training) provides an illustration of NCL. Notably, when sampling the time steps $\{t_1, \ldots, t_n\} \subset [0, 1]$, no constraints are imposed (*e.g.*, we do not enforce $t_1 > t_2$), allowing the model to experience varying degrees of noise in preceding entities, strengthening its adaptability during inference.

Algorithm 1 PyTorch Pseudo-Code for General Entity X

```
from einops import rearrange
import torch
import torch.nn.functional as F
class xAR(nn.Module):
    # Construct a sequence of entities based on the input latent.
    # Input: A continuous latent with shape (b, c, h, w).
    # Return: A sequence of entities with shape (b, s, c).
  def latent2token(self, latent):
    return latent.flatten(2).permute(0,2,1)
  def latent2cell(self, latent, k):
    # k: Group k × k spatially neighboring tokens into one cell.
    return rearrange(latent, "b c (h k1) (w k2) -> b (h w k1 k2) c", k1=k, k2=k)
  def latent2subsample(self, latent, distance):
    # distance: Group tokens based on evenly spaced distances.
    return rearrange(latent, "b c (d1 h) (d2 w) -> b (h w d1 d2) c", d1=distance, d2=distance)
  def latent2scale(self, latent, scales):
    # scales: A sequence of scale design.
    entities = [F.interpolate(latent, (i,i)).flatten(2).permute(0,2,1) for i in scales]
    entities = torch.cat(entities, dim=1)
    return entities
```

[1] We also experimented with rectangular cells (*e.g.*, cells with shape $k/2 \times 2k$ or $2k \times k/2$), but observed no significant difference compared to squared cells. Thus, we adopt the simpler squared cell design.

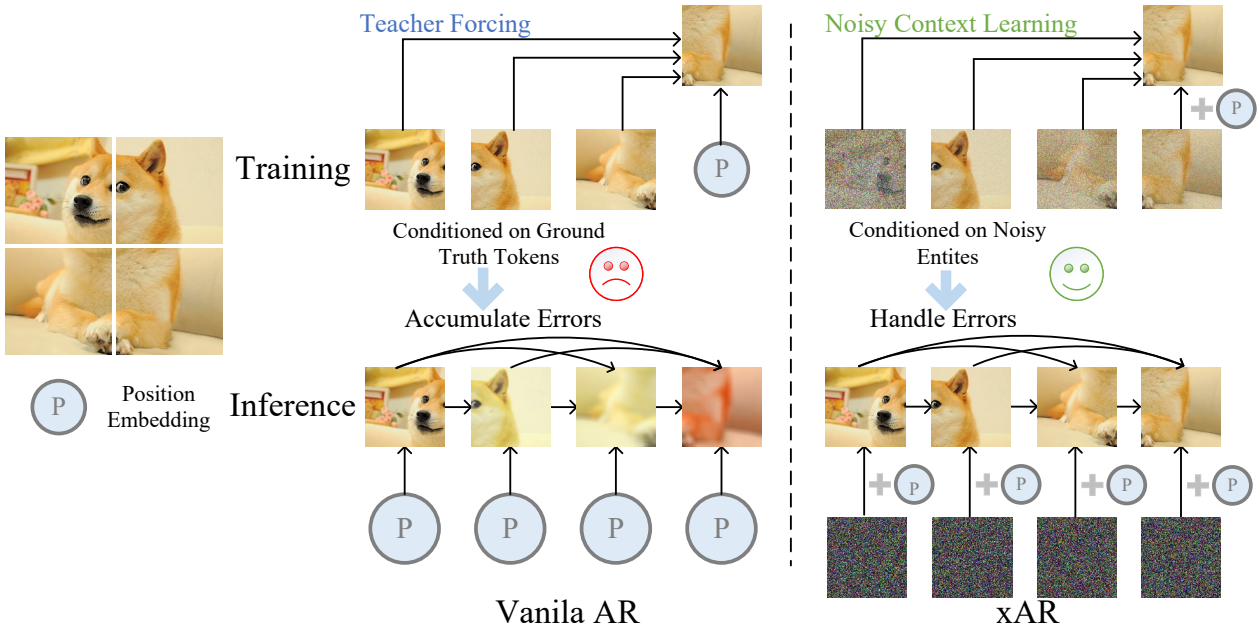

Figure 3. **Conditioning Mechanism Comparison between Vanilla AR *vs.* xAR.** During training, vanilla AR conditions on all preceding ground truth tokens (*i.e.*, Teacher Forcing), whereas xAR conditions on all previous noisy entities, each with different noises (*i.e.*, Noisy Context Learning). At inference, vanilla AR suffers from exposure bias, as errors accumulate over AR steps due to its exclusive training on ground truth tokens, leaving it unprepared for imperfect predictions. In contrast, xAR, trained to handle noisy inputs, reduces reliance on ground truth signals and improves robustness to prediction errors.

3.2.3. Inference Scheme

xAR performs autoregressive prediction at the level of entity X. Since "cell" is the default choice for X, we use it as a concrete example. As illustrated in Fig. 3 (Inference), xAR begins by predicting an initial cell $\hat{X}_1$ from a Gaussian noise sample $\epsilon_1 \sim \mathcal{N}(0, I)$ (where ϵ_1 has the same shape as $\hat{X}_1$) via flow matching [28, 30]. Conditioned on the clean estimate $\hat{X}_1$, xAR generates the next cell $\hat{X}_2$ from another Gaussian noise sample ϵ_2. This process continues autoregressively, where at the i-th AR step, the model predicts the next cell $\hat{X}_i$ based on all previously generated clean cells $\{\hat{X}_1, \cdots, \hat{X}_{i-1}\}$ and the newly drawn Gaussian noise sample ϵ_i. This iterative approach progressively refines the image, ensuring structured and context-aware generation at the cell level.

3.2.4. Discussion

As discussed in Sec. 3.1, traditional AR modeling for visual generation faces two key challenges: information density and accumulated errors. The proposed xAR is designed to address these limitations.

Semantic-Rich Prediction Entity. A cell (*i.e.*, a $k \times k$ grouping of spatially contiguous tokens) aggregates neighboring tokens, effectively capturing both local structures (*e.g.*, edges, textures) and regional contexts (*e.g.*, small objects or parts of larger objects). This leads to richer semantic representations compared to single-token predictions. By modeling relationships within the cell, the model learns to generate coherent local and regional features, shifting from isolated token-level predictions to holistic patterns. Additionally, predicting a cell rather than an individual token allows the model to reason at a higher abstraction level, akin to how NLP models predict words instead of characters. The larger receptive field per prediction step contributes more semantic information, bridging the gap between low-level visual patches and high-level semantics.

Robustness to Previous Prediction Errors. The Noisy Context Learning (NCL) strategy trains the model on noisy entities instead of perfect ground truth inputs, reducing over-reliance on pristine contexts. This alignment between training and inference distributions enhances the model's ability to handle errors in self-generated predictions. By conditioning on imperfect contexts, xAR learns to tolerate minor inaccuracies, preventing small errors from compounding into cascading errors. Additionally, exposure to noisy inputs encourages smoother representation learning, leading to more stable and consistent generations.

4. Experimental Results

In this section, we present the main results in Sec. 4.1, followed by ablation studies on key design choices in Sec. 4.2.

type	model	#params	FID↓	IS↑	Precision↑	Recall↑
GAN	BigGAN [4]	112M	6.95	224.5	0.89	0.38
GAN	GigaGAN [22]	569M	3.45	225.5	0.84	0.61
GAN	StyleGan-XL [50]	166M	2.30	265.1	0.78	0.53
Diffusion	ADM [10]	554M	10.94	101.0	0.69	0.63
Diffusion	LDM-4-G [47]	400M	3.60	247.7	-	-
Diffusion	Simple-Diffusion [20]	2B	2.44	256.3	-	-
Diffusion	DiT-XL/2 [36]	675M	2.27	278.2	0.83	0.57
Diffusion	L-DiT-3B [1]	3.0B	2.10	304.4	0.82	0.60
Diffusion	DiMR-G/2R [29]	1.1B	1.63	292.5	0.79	0.63
Diffusion	MDTv2-XL/2 [13]	676M	1.58	314.7	0.79	0.65
Diffusion	CausalFusion-H† [8]	1B	1.57	-	-	-
Flow-Matching	SiT-XL/2 [32]	675M	2.06	277.5	0.83	0.59
Flow-Matching	REPA [70]	675M	1.80	284.0	0.81	0.61
Flow-Matching	REPA† [70]	675M	1.42	305.7	0.80	0.65
Mask.	MaskGIT [6]	227M	6.18	182.1	0.80	0.51
Mask.	TiTok-S-128 [68]	287M	1.97	281.8	-	-
Mask.	MAGVIT-v2 [67]	307M	1.78	319.4	-	-
Mask.	MaskBit [62]	305M	1.52	328.6	-	-
AR	VQVAE-2 [39]	13.5B	31.11	∼45	0.36	0.57
AR	VQGAN [11]	227M	18.65	80.4	0.78	0.26
AR	VQGAN [11]	1.4B	15.78	74.3	-	-
AR	RQTran. [26]	3.8B	7.55	134.0	-	-
AR	ViTVQ [65]	1.7B	4.17	175.1	-	-
AR	DART-AR [15]	812M	3.98	256.8	-	-
AR	MonoFormer [71]	1.1B	2.57	272.6	0.84	0.56
AR	Open-MAGVIT2-XL [31]	1.5B	2.33	271.8	0.84	0.54
AR	LlamaGen-3B [54]	3.1B	2.18	263.3	0.81	0.58
AR	FlowAR-H [44]	1.9B	1.65	296.5	0.83	0.60
AR	RAR-XXL [69]	1.5B	1.48	326.0	0.80	0.63
MAR	MAR-B [27]	208M	2.31	281.7	0.82	0.57
MAR	MAR-L [27]	479M	1.78	296.0	0.81	0.60
MAR	MAR-H [27]	943M	1.55	303.7	0.81	0.62
VAR	VAR-$d16$ [55]	310M	3.30	274.4	0.84	0.51
VAR	VAR-$d20$ [55]	600M	2.57	302.6	0.83	0.56
VAR	VAR-$d30$ [55]	2.0B	1.97	323.1	0.82	0.59
xAR	xAR-B	172M	1.72	280.4	0.82	0.59
xAR	xAR-L	608M	1.28	292.5	0.82	0.62
xAR	xAR-H	1.1B	1.24	301.6	0.83	0.64

Table 1. **Generation Results on ImageNet-256.** Metrics include Fréchet Inception Distance (FID), Inception Score (IS), Precision, and Recall. † denotes the use of guidance interval sampling [25]. The proposed xAR-H achieves a state-of-the-art 1.24 FID on the ImageNet-256 benchmark without relying on vision foundation models (*e.g.*, DINOv2 [35]) or guidance interval sampling [25], as used in REPA [70].

4.1. Main Results

We conduct experiments on ImageNet [9] at 256×256 and 512×512 resolutions. Following prior works [27, 36], we evaluate model performance using FID [18], Inception Score (IS) [49], Precision, and Recall. xAR is trained with the same hyper-parameters as [27, 36] (*e.g.*, 800 training epochs), with model sizes ranging from 172M to 1.1B parameters. See Appendix A for hyper-parameter details.

ImageNet-256. In Tab. 1, we compare xAR with previous state-of-the-art generative models. Out best variant, xAR-H, achieves a new state-of-the-art-performance of 1.24 FID, outperforming the GAN-based StyleGAN-XL [50] by 1.06 FID, masked-prediction-based MaskBit [6]

model	#params	FID↓	IS↑
VQGAN [11]	227M	26.52	66.8
BigGAN [4]	158M	8.43	177.9
MaskGiT [6]	227M	7.32	156.0
DiT-XL/2 [36]	675M	3.04	240.8
DiMR-XL/3R [29]	525M	2.89	289.8
VAR-d36 [55]	2.3B	2.63	303.2
REPA‡ [70]	675M	2.08	274.6
xAR-L	608M	1.70	281.5

Table 2. **Generation Results on ImageNet-512.** ‡ denotes the use of DINOv2 [35].

by 0.28 FID, AR-based RAR [69] by 0.24 FID, VAR [55] by 0.73 FID, MAR [27] by 0.31 FID, and flow-matching-based REPA [70] by 0.18 FID. Notably, xAR does not rely on vision foundation models [35] or guidance interval sampling [25], both of which were used in REPA [70], the previous best-performing model. Additionally, our lightweight xAR-B (172M), surpasses DiT-XL (675M) [36] by 0.55 FID while achieving an inference speed of 9.8 images per second—20× faster than DiT-XL (0.5 images per second). Detailed speed comparison can be found in Appendix B.

ImageNet-512. In Tab. 2, we report the performance of xAR on ImageNet-512. Similarly, xAR-L sets a new state-of-the-art FID of 1.70, outperforming the diffusion based DiT-XL/2 [36] and DiMR-XL/3R [29] by a large margin of 1.34 and 1.19 FID, respectively. Additionally, xAR-L also surpasses the previous best autoregressive model VAR-d36 [55] and flow-matching-based REPA [70] by 0.93 and 0.38 FID, respectively.

Qualitative Results. Fig. 4 presents samples generated by xAR (trained on ImageNet) at 512×512 and 256×256 resolutions. These results highlight xAR's ability to produce high-fidelity images with exceptional visual quality.

4.2. Ablation Studies

In this section, we conduct ablation studies using xAR-B, trained for 400 epochs to efficiently iterate on model design.

Prediction Entity X. The proposed xAR extends next-token prediction to next-X prediction. In Tab. 3, we evaluate different designs for the prediction entity X, including an individual patch token, a cell (a group of surrounding tokens), a subsample (a non-local grouping), a scale (coarse-to-fine resolution), and an entire image.

Among these variants, cell-based xAR achieves the best performance, with an FID of 2.48, outperforming the token-based xAR by 1.03 FID and surpassing the second best design (scale-based xAR) by 0.42 FID. Furthermore, even when using standard prediction entities such as tokens, subsamples, images, or scales, xAR consistently outperforms existing methods while requiring significantly fewer parameters. These results highlight the efficiency and effectiveness of xAR across diverse prediction entities.

model	prediction entity	#params	FID↓	IS↑
LlamaGen-L [54]	token	343M	3.80	248.3
xAR-B		172M	3.51	251.4
PAR-L [61]	subsample	343M	3.76	218.9
xAR-B		172M	3.58	231.5
DiT-L/2 [36]	image	458M	5.02	167.2
xAR-B		172M	3.13	253.4
VAR-$d16$ [55]	scale	310M	3.30	274.4
xAR-B		172M	2.90	262.8
xAR-B	cell	172M	2.48	269.2

Table 3. **Ablation on Prediction Entity X.** Using cells as the prediction entity outperforms alternatives such as tokens or entire images. Additionally, under the same prediction entity, xAR surpasses previous methods, demonstrating its effectiveness across different prediction granularities. xAR-B is trained 400 epochs.

cell size ($k \times k$ tokens)	$m \times m$ grid	FID↓	IS↑
1×1	16×16	3.51	251.4
2×2	8×8	3.04	253.5
4×4	4×4	2.61	258.2
8×8	2×2	2.48	269.2
16×16	1×1	3.13	253.4

Table 4. **Ablation on the cell size.** In this study, a 16×16 continuous latent representation is partitioned into an $m \times m$ grid, where each cell consits of $k \times k$ neighboring tokens. A cell size of 8×8 achieves the best performance, striking an optimal balance between local structure and global context. Settings: xAR-B, 400 epochs.

Cell Size. A prediction entity cell is formed by grouping spatially adjacent $k \times k$ tokens, where a larger cell size incorporates more tokens and thus captures a broader context within a single prediction step. For a 256×256 input image, the encoded continuous latent representation has a spatial resolution of 16×16. Given this, the image can be partitioned into an $m \times m$ grid, where each cell consists of $k \times k$ neighboring tokens. As shown in Tab. 4, we evaluate different cell sizes with $k \in \{1, 2, 4, 8, 16\}$, where $k = 1$ represents a single token and $k = 16$ corresponds to the entire image as a single entity. We observe that performance improves as k increases, peaking at an FID of 2.48 when using cell size 8×8 (*i.e.*, $k = 8$). Beyond this, performance declines, reaching an FID of 3.13 when the entire image is treated as a single entity. These results suggest that using cells rather than the entire image as the prediction unit allows the model to condition on previously generated context, improving confidence in predictions while maintaining both rich semantics and local details.

Noisy Context Learning. During training, xAR employs Noisy Context Learning (NCL), predicting X_n by condi-

Figure 4. **Generated Samples.** xAR generates high-quality images at resolutions of 512×512 (1st row) and 256×256 (2nd and 3rd row).

previous cell	noise time step	FID↓	IS↑
clean	$t_i = 0, \forall i < n$	3.45	243.5
increasing noise	$t_1 < t_2 < \cdots < t_{n-1}$	2.95	258.8
decreasing noise	$t_1 > t_2 > \cdots > t_{n-1}$	2.78	262.1
random noise	no constraint	2.48	269.2

Table 5. **Ablation on Noisy Context Learning.** This study examines the impact of noise time steps ($t_1, \cdots, t_{n-1} \subset [0, 1]$) in previous entities ($t = 0$ represents pure Gaussian noise). Conditioning on all clean entities (the "clean" variant) results in suboptimal performance. Imposing an order on noise time steps, either "increasing noise" or "decreasing noise", also leads to inferior results. The best performance is achieved with the "random noise" setting, where no constraints are imposed on noise time steps. Settings: xAR-B, 400 epochs.

tioning on all previous noisy entities, unlike Teacher Forcing. The noise intensity of previous entities is contorlled by noise time steps $\{t_1, \ldots, t_{n-1}\} \subset [0, 1]$, where $t = 0$ corresponds to pure Gaussian noise. We analyze the impact of NCL in Tab. 5. When conditioning on all clean entities (*i.e.*, the "clean" variant, where $t_i = 0, \forall i < n$), which is equivalent to vanilla AR (*i.e.*, Teacher Forcing), the suboptimal performance is obtained. We also evaluate two constrained noise schedules: the "increasing noise" variant, where noise time steps increase over AR steps ($t_1 < t_2 < \cdots < t_{n-1}$), and the " decreasing noise" variant, where noise time steps decrease ($t_1 > t_2 > \cdots > t_{n-1}$). While both settings improve over the "clean" variant, they remain inferior to our final "random noise" setting, where no constraints are imposed on noise time steps, leading to the best performance.

5. Conclusion

In this work, we introduced xAR, a general next-X prediction framework for autoregressive visual generation. Unlike traditional next-token prediction, xAR reformulates discrete token classification as continuous entity regression, enabling more flexible and semantically meaningful prediction units. Through systematic exploration, we found that next-cell prediction provides the best balance between local structure and global coherence. To mitigate exposure bias, we proposed Noisy Context Learning (NCL), which trains the model on noisy entities instead of pristine ground truth inputs, improving robustness and reducing cascading errors. As a result, xAR achieves state-of-the-art performance on ImageNet-256 and ImageNet-512.

Acknowledge

This work is supported by ONR N00014-21-1-2690 and ONR award N000142412696.

References

[1] Alpha-vllm. large-dit-imagenet. 2024. 6

[2] Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Chi Kit Cheung. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, 2022. 3

[3] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *CVPR*, 2023. 3

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 6, 7

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020. 2

[6] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *CVPR*, 2022. 6, 7

[7] Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020. 2

[8] Chaorui Deng, Deyao Zh, Kunchang Li, Shi Guan, and Haoqi Fan. Causal diffusion transformers for generative modeling. *arXiv preprint arXiv:2412.12095*, 2024. 6

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2, 6

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 34, 2021. 6

[11] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 1, 2, 3, 4, 6, 7

[12] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 3

[13] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Mdtv2: Masked diffusion transformer is a strong image synthesizer. *arXiv preprint arXiv:2303.14389*, 2023. 6

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 2014. 3

[15] Jiatao Gu, Yuyang Wang, Yizhe Zhang, Qihang Zhang, Dinghuai Zhang, Navdeep Jaitly, Josh Susskind, and Shuangfei Zhai. Dart: Denoising autoregressive transformer for scalable text-to-image generation. In *ICLR*, 2025. 6

[16] Ju He, Qihang Yu, Qihao Liu, and Liang-Chieh Chen. Flowtok: Flowing seamlessly across text and image tokens. In *ICCV*, 2025. 3

[17] Tianxing He, Jingzhao Zhang, Zhiming Zhou, and James Glass. Exposure bias versus self-recovery: Are distortions really incremental for autoregressive text generation? In *EMNLP*, 2021. 3

[18] 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. *NeurIPS*, 30, 2017. 6

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 3

[20] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *ICML*, 2023. 3, 6

[21] Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion (sid2): 1.5 fid on imagenet512 with pixel-space diffusion. *arXiv preprint arXiv:2410.19324*, 2024. 3

[22] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *CVPR*, 2023. 6

[23] Dongwon Kim, Ju He, Qihang Yu, Chenglin Yang, Xiaohui Shen, Suha Kwak, and Liang-Chieh Chen. Democratizing text-to-image masked generative models with compact text-aware one-dimensional tokens. In *ICCV*, 2025. 2

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 3

[25] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *arXiv preprint arXiv:2404.07724*, 2024. 2, 6, 7

[26] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *CVPR*, 2022. 6

[27] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *NeurIPS*, 2024. 2, 3, 4, 6, 7

[28] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 1, 2, 3, 4, 5

[29] Qihao Liu, Zhanpeng Zeng, Ju He, Qihang Yu, Xiaohui Shen, and Liang-Chieh Chen. Alleviating distortion in image generation via multi-resolution diffusion models. *NeurIPS*, 2024. 3, 6, 7

[30] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 2, 3, 4, 5

[31] Zhuoyan Luo, Fengyuan Shi, Yixiao Ge, Yujiu Yang, Limin Wang, and Ying Shan. Open-magvit2: An open-source project toward democratizing auto-regressive visual generation. *arXiv preprint arXiv:2409.04410*, 2024. 6

[32] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *ECCV*, 2024. 2, 3, 6

[33] OpenAI. Introducing chatgpt. https://openai.com/blog/chatgpt/, 2022. 2

[34] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[35] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 3, 6, 7

[36] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 2, 3, 6, 7

[37] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf, 2018. 2

[38] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *ICLR*, 2016. 2, 4

[39] Ali Razavi, Aaron Van Den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *NeurIPS*, 2019. 2, 6

[40] Jingjing Ren, Wenbo Li, Haoyu Chen, Renjing Pei, Bin Shao, Yong Guo, Long Peng, Fenglong Song, and Lei Zhu. Ultrapixel: Advancing ultra high-resolution image synthesis to new peaks. *NeurIPS*, 2024. 3

[41] Jingjing Ren, Wenbo Li, Zhongdao Wang, Haoze Sun, Bangzhen Liu, Haoyu Chen, Jiaqi Xu, Aoxue Li, Shifeng Zhang, Bin Shao, et al. Turbo2k: Towards ultra-efficient and high-quality 2k video synthesis. *arXiv preprint arXiv:2504.14470*, 2025. 3

[42] Sucheng Ren, Yaodong Yu, Nataniel Ruiz, Feng Wang, Alan Yuille, and Cihang Xie. M-var: Decoupled scale-wise autoregressive modeling for high-quality image generation. *arXiv preprint arXiv:2411.10433*, 2024. 2

[43] Sucheng Ren, Xianhang Li, Haoqin Tu, Feng Wang, Fangxun Shu, Lei Zhang, Jieru Mei, Linjie Yang, Peng Wang, Heng Wang, et al. Autoregressive pretraining with mamba in vision. In *ICLR*, 2025.

[44] Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Flowar: Scale-wise autoregressive image generation meets flow matching. In *ICML*, 2025. 2, 6

[45] Sucheng Ren, Qihang Yu, Ju He, Alan Yuille, and Liang-Chieh Chen. Grouping first, attending smartly: Training-free acceleration for diffusion transformers. *arXiv preprint arXiv:2505.14687*, 2025. 3

[46] Sucheng Ren, Hongru Zhu, Chen Wei, Yijiang Li, Alan Yuille, and Cihang Xie. Arvideo: Autoregressive pretraining for self-supervised video representation learning. *Transactions on Machine Learning Research*, 2025. 2

[47] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3, 4, 6

[48] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 3

[49] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *NeurIPS*, 29, 2016. 6

[50] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. *arXiv preprint arXiv:2201.00273*, 2022. 3, 6

[51] Inkyu Shin, Chenglin Yang, and Liang-Chieh Chen. Deeply supervised flow-based generative models. In *ICCV*, 2025. 3

[52] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3

[53] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 2019. 3

[54] 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. 2, 3, 6, 7

[55] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *NeurIPS*, 2024. 1, 2, 3, 4, 6, 7

[56] 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. 2

[57] Aaron Van Den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *NeurIPS*, 2016. 2

[58] Aaron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, 2016. 2

[59] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *NeurIPS*, 2017. 2, 3

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 2, 3

[61] Yuqing Wang, Shuhuai Ren, Zhijie Lin, Yujin Han, Haoyuan Guo, Zhenheng Yang, Difan Zou, Jiashi Feng, and Xihui Liu. Parallelized autoregressive visual generation. *arXiv preprint arXiv:2412.15119*, 2024. 7

[62] Mark Weber, Lijun Yu, Qihang Yu, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. Maskbit: Embedding-free image generation via bit tokens. *arXiv preprint arXiv:2409.16211*, 2024. 6

[63] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989. 2, 3

[64] Chenglin Yang, Celong Liu, Xueqing Deng, Dongwon Kim, Xing Mei, Xiaohui Shen, and Liang-Chieh Chen. 1.58-bit flux. *arXiv preprint arXiv:2412.18653*, 2024. 3

[65] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge,

and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021. 6

[66] 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. 2, 3

[67] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion–tokenizer is key to visual generation. In *ICLR*, 2024. 6

[68] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. *NeurIPS*, 2024. 6

[69] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. In *ICCV*, 2025. 2, 6, 7

[70] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024. 2, 3, 6, 7

[71] Chuyang Zhao, Yuxing Song, Wenhao Wang, Haocheng Feng, Errui Ding, Yifan Sun, Xinyan Xiao, and Jingdong Wang. Monoformer: One transformer for both diffusion and autoregression. *arXiv preprint arXiv:2409.16280*, 2024. 6