Spanning Tree Autoregressive Visual Generation

Sangkyu Lee[1,2,⋆], Changho Lee[3], Janghoon Han[3], Hosung Song[3], Tackgeun You[2], Hwasup Lim[2], Stanley Jungkyu Choi[3], Honglak Lee[3,4], and Youngjae Yu[5,†]

[1] Yonsei University, Seoul, Korea
[2] Korea Institute of Science and Technology, Seoul, Korea
[3] LG AI Research, Seoul, Korea
[4] University of Michigan, Ann Arbor, USA
[5] Seoul National University, Seoul, Korea

Abstract. We present Spanning Tree Autoregressive (STAR) modeling, which can incorporate prior knowledge of images, such as center bias and locality, to maintain sampling performance while also providing sufficiently flexible sequence orders to accommodate image editing at inference time. Approaches that expose conventional autoregressive (AR) models in visual generation to arbitrary sequence orders via random permutation suffer from degraded sampling performance or compromise the flexibility in sequence order choice at inference time. Instead, STAR utilizes traversal orders of uniform spanning trees in a lattice defined by the positions of image patches. Traversal orders are obtained via breadth-first search, allowing us to efficiently construct a spanning tree via rejection sampling whose traversal order ensures that the connected partial observation of the image appears as a prefix for native image inpainting support. Through the tailored yet structured sequence order randomization strategy, STAR preserves the capability of postfix completion while maintaining sampling performance, without any significant changes to the model architecture widely adopted in language AR modeling.[‡]

Keywords: Generative Model · Image Generation · Image Editing

1 Introduction

The scalability of the *next-token prediction* pretraining scheme with autoregressive (AR) models built on a Decoder-only Transformer architecture [15, 21, 35] has yielded remarkable results in language generation, represented as large language models [35, 45]. Meanwhile, the success of the Decoder-only Transformer has led to the transition of AR modeling for visual generation [47, 48] to the *next-patch prediction* scheme, which typically predicts the next token representing a single patch by the predefined *raster-scan* order. Similar to the language

⋆ Work mainly done during internship at LG AI Research.
† Corresponding author
‡ The code is available at `https://github.com/oddqueue/star`.

Fig. 1: (a) Conventional AR visual generation models follow a fixed raster-scan order, limiting the flexibility of inference sequence order. (b) Random permutation offers flexibility in the inference sequence order, but does not reflect prior knowledge, such as center bias and locality. (c) STAR adopts the traversal order of the uniform spanning tree, thereby structurally maintaining prior knowledge and flexibility at inference time.

generation, AR modeling for visual generation through the next-token prediction scheme also exhibits efficient scaling behavior [12, 22, 28], highlighting the competitive advantage of AR visual modeling compared to other methods for visual generation [2, 6, 18, 26, 34]. Furthermore, the architecture that seamlessly integrates with language generation serves as a cornerstone of early-fusion multimodal models [43] for native support for visual and language generation.

However, conventional AR visual modeling restricts the 2D nature of images to a specific unidirectional dependency defined by the fixed sequence order. To address the unidirectional problem of AR models in visual generation, recent approaches [26, 32, 54] adopt the *random permutation* [7, 33, 52] to shuffle the sequence order of tokens and provide next-patch positions to be predicted as additional conditions to expose AR models to the diverse sequence order during training. Nevertheless, AR models trained with arbitrary sequence orders tend to exhibit degraded performance compared to conventional AR models that follow a fixed raster-scan order, despite having flexibility in choosing inference sequence orders required for native region-level modification capabilities to achieve image editing while maintaining the Decoder-only Transformer architecture.

In this paper, we present **Spanning Tree Autoregressive (STAR)** modeling, a structured sequence order randomization strategy based on the *uniform spanning tree* for AR visual generation models by revisiting the characteristic of raster-scan order and random permutation, as described in Figure 1. Training solely in raster-scan order forces the choice of sequence order at inference time. However, it guarantees continuity in the 2D lattice between the tokens in the prefix and postfix. Also, it starts from the corner where salient objects are unlikely to appear. On the other hand, random permutation does not guarantee this property of continuity and starting from the corner, but it does guarantee diverse sequence orders due to randomness. This is the motivation of our approach, which traverses uniform spanning trees, as a representative structure that satisfies the advantages of both, reflecting *prior knowledge* of images, center bias [4, 41] and locality [3, 20], yet preserving *randomness* for flexibility.

Specifically, we consider the traversal order obtained by breadth-first search (BFS) of the uniform sampling tree, with a random corner as the root node, as

the sequence order. By choosing the BFS traversal order, we can efficiently sample a spanning tree that can handle image inpainting as a postfix completion via rejection sampling at inference time, when the given partial observation of the image is connected in the lattice, unlike the conventional raster-scan order AR visual generation model. Furthermore, STAR requires minimal changes to the conventional Decoder-only Transformer, only adding additional features to provide positional information that condition the next-token position [33, 54]. This feature is shared with the random permutation strategy, but STAR preserves image generation performance while still supporting image inpainting.

The contribution of our work, STAR, can be summarized as threefold:

- We propose a structured sequence order randomization strategy for AR visual generation based on the BFS traversal of the uniform sampling tree, ensuring both flexibility of sequence order and sampling performance.
- We show that the BFS traversal order of uniform spanning trees is sufficient to achieve the benefits of random permutation for diverse sequence orders in AR visual generation models without increasing training complexity.
- We introduce a new building block that supports region-level modification capabilities for image editing with minimal changes that do not sacrifice performance, a challenge in conventional AR visual generative models.

2 Related Work

2.1 Unidirectional Autoregressive Visual Generation

Decoder-only Transformer proposed by GPT-2 [35], which adopts the causal attention mask, has become the de facto standard architecture for AR language modeling [5, 45] due to its advantageous feature of scaling behavior [15, 21]. The success of the Decoder-only Transformer, often represented by next-token prediction, has enriched the research direction of image generation based on AR modeling [47, 48] to next-token prediction. A sequence of research for next-token prediction for image generation [13, 40, 53] achieves AR visual modeling by training models to predict patch representations of images, typically arranged in a predefined, unidirectional order. Here, a series of works [12, 13, 54] commonly observe that raster-scan order outperforms variants of sequence order, such as space-filling curves, for the predefined unidirectional order, and raster-scan is widely adopted as the standard sequence order for AR visual modeling. Sharing the architecture across data modalities facilitates expansion into early-fusion multimodal models [43], but unidirectional modeling of images imposes limitations on image editing due to their 2D structure. Inspired by this aspect, we aim to enable flexible downstream task capabilities for AR visual generation models without significant architectural changes from the Decoder-only Transformer.

2.2 Bidirectional Modeling for Visual Generation

A line of research has been conducted to address the bidirectional dependency between image patches in visual generation. One research direction is to directly

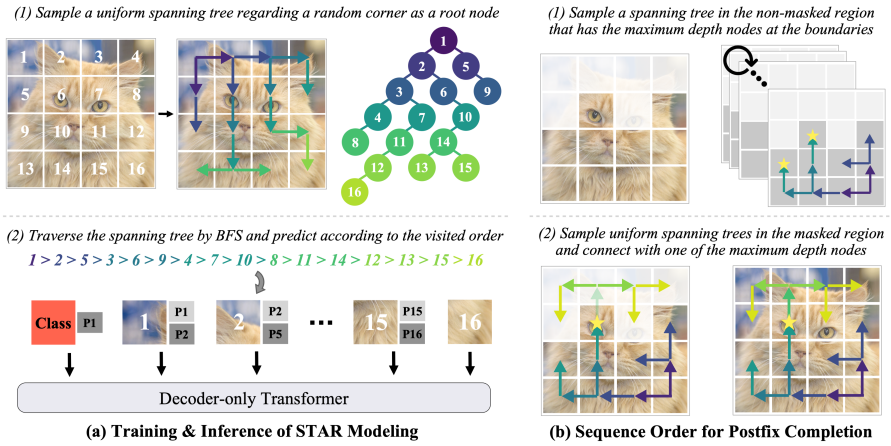


Fig. 2: Overview of STAR. (a) We perform training and inference in the sequence order of a BFS traversal for uniform spanning trees that start at the corners, additionally conditioning on the next-token positions. (b) We perform rejection sampling of the spanning tree with the maximum depth at the boundaries of the non-masked region, allowing postfix completion of partial observations to achieve region-level modification.

introduce bidirectional transformers for masked token prediction [6], diffusion models [2, 18, 34], or to mimic AR models with diffusion loss [26]. However, they often show less efficiency than the AR models in scaling behavior [22, 28]. Another approach decouples spatial prediction in AR visual modeling and introduces a dedicated transformer architecture [25, 29, 36, 37, 44], but this can limit the simplicity of extension to early-fusion multimodal models. The other approach exposes AR models by applying random permutation to the sequence order [7] and providing the conditions for next-token position through position instruction tokens [32], additional positional embeddings [33,54], or multi-stream attention mechanisms [26,52], but commonly trade either sampling performance or flexibility in sequence order at inference time. As an extension of the last approach, we investigate how to extend the trade-off frontier between performance and flexibility while preserving the Decoder-only Transformer architecture.

3 Method

In this section, we describe our approach, **Spanning Tree Autoregressive (STAR)** modeling, an autoregressive visual modeling based on the traversal order of the uniform spanning tree to impose structurally randomized sequence orders preserving both of image generation performance and flexibility in choice of sequence orders to accommodate image editing, as described in Figure 2.

3.1 From Fixed To Randomized Sequence Orders

As generative models for visual generation, AR models approximate the marginal distribution of the image x by factorizing it into conditional distributions trained

with maximum likelihood estimation. Here, the conventional approach treats an image as a collection of patch token representations $\{x_{i,j}\}$ of size $h \times w$. These 2D-structured tokens are linearized to the 1D sequence order $\tau \coloneqq (\tau_1, \cdots \tau_N) : \mathbb{Z}_h \times \mathbb{Z}_w \to \mathbb{Z}_N$, which serves as a prediction order for AR models where $N = h \times w$. Formally, AR models p_θ for visual generation are trained as follows:

$$\underset{\theta}{\operatorname{argmax}}\ p_\theta(x) = p_\theta(x_{\tau_0}) \prod_{i=1}^{N} p_\theta(x_{\tau_i} | x_{\tau_1}, x_{\tau_2}, \cdots x_{\tau_{i-1}}) = \prod_{i=0}^{N} p_\theta(x_{\tau_i} | x_{\tau_{<i}}). \quad (1)$$

The choice of sequence order τ is not a straightforward feature in visual generation compared to natural language generation. It is challenging to impose an adequate sequence order on images because of their 2D structure, unlike natural languages, which exhibit a relatively straightforward left-to-right sequence order. The empirically practical approach is to adopt the well-known sequence order, specifically the raster-scan order, as it often demonstrates better performance compared to other sequence orders [12, 13, 54]. However, adopting a single fixed sequence order imposes a structural limitation on the AR models: it exposes the image to the model only in a single, unidirectional sequence. In other words, it prevents conditioning on tokens that do not appear as prefixes in the predefined sequence order, unlike bidirectional modeling for visual generation [6, 34].

In this context, some approaches uniformly sample multiple sequence orders from random permutations S_N during training, rather than fixing to a single order [7, 26, 27, 32, 54]. The common goal is to enable AR models to be trained on randomly ordered contexts as a bidirectional modeling in *expectation*:

$$\underset{\theta}{\operatorname{argmax}}\ p_\theta(x) = \mathbb{E}_{\tau \sim S_N} \Big[\prod_{i=0}^{N} p_\theta(x_{\tau_i} | x_{\tau_{<i}}) \Big]. \quad (2)$$

This approach has been investigated under the name of *permutation* AR modeling [33, 52] beyond visual generation. Meanwhile, it is known that this loss objective is theoretically equivalent to the loss objective for masked token prediction [31, 46], and the difference between the two approaches is whether they use the Decoder-only Transformer as the model architecture or not [51]. Still, this flexibility in sequence order adds a key downstream feature for visual generation: native support of region-level modification required for image editing. Given the 2D position indices of non-masked patches $\hat{I} \times \hat{J}$, we can achieve image inpainting by simply finding a sequence order $\hat{\tau} \coloneqq (\tau_{\text{pre}}, \tau_{\text{post}})$ that ensures the partial observation of the image appears as a prefix for completing the postfix:

$$\tau_{\text{pre}} : \hat{I} \times \hat{J} \to \mathbb{Z}_{|\hat{I} \times \hat{J}|},\ \tau_{\text{post}} : (\mathbb{Z}_h \times \mathbb{Z}_w) \setminus (\hat{I} \times \hat{J}) \to |\hat{I} \times \hat{J}| + \mathbb{Z}_{|(\mathbb{Z}_h \times \mathbb{Z}_w) \setminus (\hat{I} \times \hat{J})|}. \quad (3)$$

However, this permutation AR modeling often converges more slowly than conventional AR modeling that uses a single predefined sequence known to be advantageous for the intrinsic structure [51]. In theory, if we could perfectly model the data distribution, the model entropy $\mathcal{H}_\theta(x)$ should converge to the data entropy $\mathcal{H}(x)$ for any permutation because of the chain rule for entropy

(i.e $\mathcal{H}(x) \approx \sum_{i=0}^{N} \mathcal{H}_\theta(x_{\tau_i}|x_{\tau_{<i}})\ \forall \tau \in S_N$), but empirically this does not happens because of the different convergence speed across sequence orders. The existence of advantageous sequence orders in convergence speed, such as the raster-scan order, has been repeatedly observed in previous work [12, 13, 54]. Nevertheless, permutation AR modeling is forced to maximize the likelihood over all sequence orders rather than the advantageous ones, leading to inferior performance compared to the conventional AR model within the same computation budget.

3.2 Uniform Spanning Tree for Autoregressive Modeling

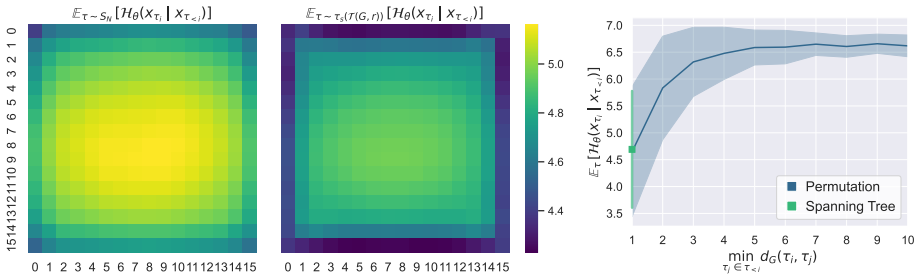
The inherent nature of permutation AR modeling underscores the need for an approach that improves convergence speed while retaining the advantage of native image inpainting capability. Nonetheless, prior work leverages permutation AR modeling only as a pretext task for conventional AR models [54] or focuses on parallel inference [26,32] as similarly investigated in masked token prediction [6]. It leads to increased training complexity due to additional hyperparameter search while sacrificing native image inpainting capability, or directly results in inferior sampling performance compared to AR models using the raster-scan order.

The native image inpainting capability can be achieved if the model supports random sequence orders that are sufficient to find a specific sequence order that makes a given partial observation appear in the prefix and a masked region appear in the postfix. However, random permutation exposes all $N!$ possible sequence orders beyond the tighter sets that support native image inpainting, including non-advantageous orders that can affect convergence speed. Therefore, we need to narrow the sample space of sequence orders by identifying the common properties underlying well-known predefined sequence orders, such as raster-scan order and space-filling curves (e.g., Hilbert and Morton curves).

In this respect, we propose an approach that uses the *uniform spanning tree* on a lattice corresponding to the token position relationships in images to structurally reflect prior knowledge about the image in the sequence order randomization strategy. Formally, we regard the 2D token position index (i, j) as a vertex and each adjacent vertex $u, v \in V$ is connected by edge $(u, v) \in E$ as a regular taxicab lattice $G \coloneqq (V, E)$. We uniformly sample a spanning tree $T \sim \mathcal{T}(G, r)$, which is a subgraph of G that forms a tree that includes all vertices of G while considering a randomly chosen corner vertex as the root node r. Then, we train the AR model p_θ by maximum likelihood estimation with additional conditioning on the next-token position according to the traversal order τ_s:

$$\underset{\theta}{\operatorname{argmax}}\ \mathbb{E}_{\tau \sim \tau_s\left(\mathcal{T}(G,r)\right)}\Big[\prod_{i=0}^{N} p_\theta(x_{\tau_i}|x_{\tau_{<i}})\Big]. \tag{4}$$

This stems from the conjecture that the advantage of predefined sequences is derived from *prior knowledge* of the image, such as *center bias* [4, 41] and *locality* [3, 20]. To verify whether similar phenomena persist in token representations of images, we measure and analyze conditional model entropy $\mathcal{H}_\theta(x_{\tau_i} \mid x_{\tau_{<i}})$ on validation images of ImageNet-1k [9] by sampling 30 sequence orders per



(a) (b)

Fig. 3: (a) Conditional model entropy according to token position. Higher token entropy in central regions indicates that token position affects the differences in prediction difficulty. (b) Conditional model entropy according to the minimum Manhattan distance from the prefix token positions. Tokens located at closer distances tend to exhibit lower entropy, suggesting that adjacent tokens are easier to predict than others.

single image for each AR model trained with random permutation and uniform spanning tree. Figure 3a and Figure 3b demonstrate that prior knowledge of the image is maintained in token representations; AR models show difficulty in predicting token positions moving away from the already predicted tokens or moving towards the central region, which is likely to include complex salient objects. In particular, the fact that the conditional model entropy increases rapidly with increasing distance from adjacent tokens suggests that partial observations cannot serve as meaningful context unless they are connected to the token being predicted, a similar observation to that found in masked token prediction [3].

However, the sequence order of traversing the uniform spanning tree from the corners naturally moves towards the central region from the corner while predicting adjacent tokens to the prefix. Furthermore, it is known that the number of spanning trees asymptotically follows $\exp(N \cdot z_G)$ as $N \to \infty$ where z_G is a constant defined by the lattice type, $z_G \approx 1.166$ for our regular taxicab lattice [39]. Therefore, the number of spanning trees exhibits exponential growth with N, a distinctly smaller sample space compared to the $N!$ sample space of random permutation. Still, the uniform spanning tree can be efficiently sampled based on the loop-erased random walk, also known as Wilson's algorithm, requiring $O(N \log N)$ time for sampling a uniform spanning tree in the 2D lattice [50].

3.3 Breadth-first Search Traversal for Postfix Completion

The sequence order derived from traversing uniform spanning trees structurally maintains prior knowledge of images, but this does not guarantee that we can easily find a spanning tree whose traversal order accomplishes image inpainting for a given arbitrary mask. Unlike permutation AR modeling, which can simply sample the prefix sequence order τ_{pre} and the postfix sequence order τ_{post} by sampling permutations in each non-masked and masked region, we need to sample the spanning tree according to the traversal strategy. Thus, we need to choose an appropriate traversal strategy that can easily sample a spanning tree that satisfies the requirement for postfix completion at inference time.

Algorithm 1 Sampling Sequence Orders for Postfix Completion

Require: lattice $G \coloneqq (V, E)$, mask $M \coloneqq (V_M, E_M)$, corners $R \subset V \setminus V_M$
Ensure: $M \subset G$, $G \setminus M$ is connected, $R \neq \emptyset$
`# (a) Find boundaries of the non-masked region and the farthest root`
$G' \leftarrow (V \setminus V_M, \{(u,v) \in E \mid u, v \notin V_M\})$
$B \leftarrow \bigcup_i B_i = \bigcup_i \{v \in V \setminus V_{M_i} \mid \exists\, u \in V_{M_i} \text{ s.t. } (u,v) \in E\}$
$r_{G'} \leftarrow \arg\max_{r \in R} \min_{B_i \subset B} \frac{1}{|B_i|} \sum_{v \in B_i} d_{G'}(v, r)$
`# (b) Sample the spanning tree in the non-masked region by rejection`
repeat
  $T_{G'} \leftarrow (V_{G'}, E_{T_{G'}}) \sim \mathcal{T}(G', r_{G'})$
  $V_{G'_{\max}} \leftarrow \{v \in V_{T_{G'}} \mid \arg\max_{v \in V_{G'}} \text{depth}_{T_{G'}}(v, r_{G'})\}$
until $\bigwedge_i (B_i \cap V_{G'_{\max}} \neq \emptyset)$
`# (c) Sample spanning trees in the masked region and connect trees`
$E_C \leftarrow \bigcup_i E_{C_i} = \bigcup_i \big\{(u, v_i) \mid u \in B_i,\ v_i \sim \{v \in V_{M_i} \mid \exists\, u \in B_i \text{ s.t. } (u,v) \in E\}\big\}$
$T_M \leftarrow (V_M, E_{T_M}) = \bigcup_i T_{M_i} \sim \mathcal{T}(M_i, v_i)$
$T_G \leftarrow \big((G, E_{T_{G'}} \cup E_C \cup E_{T_M}), r_{G'}\big)$
return sequence order $\tau_s(T_G)$

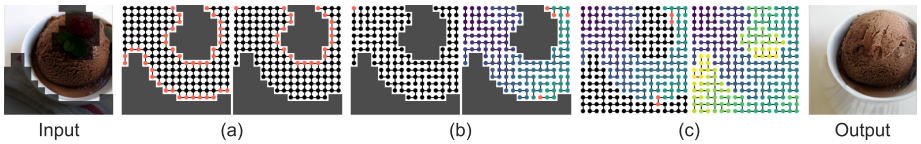


Fig. 4: Example of sampling the sequence order for postfix completion of the given input image. We chronologically illustrate the procedures (a), (b), and (c) in Algorithm 1 while highlighting r, B_i, maximum depth vertices, and spanning tree connections.

Representative approaches for traversing a tree structure include the depth-first search (DFS) and breadth-first search (BFS) algorithms. However, under a fixed tie-breaking rule, BFS has an advantage over DFS in rejection sampling to accomplish postfix completion at inference time. Formally, consider a typical image editing scenario with a given entire mask $M \coloneqq (V_M, E_M) = \bigcup_i M_i$, which is a collection of mutually disconnected masks $M_i \coloneqq (V_{M_i}, E_{M_i})$. Assuming M does not cover all corners and does not divide partial observations into disconnected graphs, the following condition must be satisfied for the spanning tree $T \sim \mathcal{T}(G, r)$ to be able to perform postfix completion in the traversal order:

$$\text{depth}_T(v_M) \geq \text{depth}_T(v) \quad \forall\, v \in V \setminus V_M,\ v_M \in V_M. \tag{5}$$

This stems from the inherent property of BFS, where the postfix vertex depth must be greater than or equal to the prefix vertex depth. In other words, if there is no vertex in the spanning tree whose depth is the same as the depth of a vertex in the non-masked region $V \setminus V_M$, then the depth of the vertex in the spanning tree must be shallower than that of all vertices belonging to M. This is equivalent to the condition that the restriction of the spanning tree $T|_{V \setminus V_M}$ has maximum depth at each boundary B_i, which is the collection of vertices in $V \setminus V_{M_i}$ adjacent to M_i. Therefore, it is sufficient to sample a spanning tree $T|_{V \setminus V_M}$ where at least one of the vertices with the maximum depth is adjacent to each M_i, then complete the spanning tree in G by sampling spanning trees in

M_i and connecting to these vertices to ensure depth increments without checking tie-breaking violations. In contrast, adopting DFS imposes a much stricter constraint even if M is given as a single connected mask, since it requires exactly one vertex visited last during the traversal of $T|_{V \setminus V_M}$ to be adjacent to M.

Thus, in our approach, we choose the traversal order τ_s as BFS during both training and inference. If partial observation is provided at inference time, we first sample a spanning tree in the non-masked region to enable postfix completion via rejection sampling, treating the corner farthest from the closest boundary in Manhattan distance as the root node to increase acceptance. We select one random vertex in the masked region adjacent to the maximum depth nodes corresponding to each B_i and treat it as the new root for the spanning tree in M_i. Finally, we complete the spanning tree in G by uniformly sampling spanning trees for M_i rooted at the selected root nodes, and connecting each root node to B_i using a single edge connection. Algorithm 1 and Figure 4 demonstrate sampling sequence order for postfix completion. Further discussion about scenarios that violate our assumption of masking is included in the supplementary material.

4 Experiments

4.1 Implementation Details

Dataset and Model Architecture We assess the feasibility of STAR by class-conditional generative models on ImageNet-1k [9] at 256×256 resolution with minimal changes to the conventional Decoder-only Transformer, following the configurations proposed by RAR [54]. We use MaskGIT-VQGAN tokenizer [6], a CNN-based VQ autoencoder trained on ImageNet-1k at 256×256 with a 1024-codebook size and a 16 downsampling rate. We adopt the ViT [11] architecture with additional features, including causal attention mask, QK LayerNorm [8], class conditioning by adaLN [34] with the first token of the sequence. We choose learnable positional embeddings for conditioning of next-token positions as in previous work [33, 54] without testing alternatives such as RoPE, which separately encodes next-token positions [16], or adaLN, which regards next-token positions as token-level conditions. We scale the model parameters into four configurations, denoted as STAR-B, STAR-L, STAR-XL, and STAR-XXL.

Training and Evaluation We apply the ten-crop transformation [42, 54] and train for 250k steps with a batch size of 2048 ($\sim$ 400 epochs) for all configurations. We use AdamW [30], cosine learning rate scheduler with linear warmup, Wilson's algorithm [50] for sampling uniform spanning trees, fixed tie-breaking rule that first visits the index preceding others in raster-scan order during the BFS, class dropout with a probability of 0.1 for classifier-free guidance [19], and power-cosine CFG scheduler [14, 49] without the top-p or top-k sampling at inference time. We follow the evaluation protocol of ADM [10] and report Fréchet Inception Distance (FID) [17], Inception Score (IS) [38], Precision, and Recall [24]. We extend this protocol to quantitatively evaluate image inpainting performance

Table 1: Comparison of class-conditional image generation on ImageNet-1k at 256 × 256. We report Fréchet Inception Distance (FID), Inception Score (IS), Precision (P), and Recall (R), with ↓ or ↑ for indicating lower or higher is better in the given metric.

Tokenizer	Model Type	Model	# Params	FID(↓)	IS(↑)	P(↑)	R(↑)
VAE	Diffusion	UViT-L/2 [2]	287M	3.40	219.9	0.83	0.52
		UViT-XL/2 [2]	501M	2.29	263.9	0.82	0.57
		DiT-L/2 [34]	458M	5.02	167.2	0.75	0.57
		DiT-XL/2 [34]	675M	2.27	278.2	0.83	0.57
	Bidirectional AR	MAR-B [27]	208M	2.31	281.7	0.82	0.57
		MAR-L [27]	479M	1.78	296.0	0.81	0.60
		MAR-H [27]	943M	1.55	303.7	0.81	0.62
VQ	Raster-scan AR	LlamaGen-L [40]	343M	3.80	248.3	0.83	0.51
		LlamaGen-XL [40]	804M	3.39	227.1	0.81	0.54
		LlamaGen-XXL [40]	1.4B	3.09	253.6	0.83	0.53
		LlamaGen-3B [40]	3.0B	3.05	222.3	0.80	0.58
		RAR-B [54]	261M	1.95	290.5	0.82	0.58
		RAR-L [54]	461M	1.70	299.5	0.81	0.60
		RAR-XL [54]	955M	1.50	306.9	0.80	0.62
		RAR-XXL [54]	1.5B	1.48	326.0	0.80	0.63
	Block-wise AR	VAR-*d*16 [44]	310M	3.30	274.4	0.84	0.51
		VAR-*d*20 [44]	600M	2.57	302.6	0.83	0.56
		VAR-*d*24 [44]	1.0B	2.09	312.9	0.82	0.59
		VAR-*d*30 [44]	2.0B	1.92	323.1	0.82	0.59
	Randomized AR	RandAR-L [32]	343M	2.55	288.8	0.81	0.58
		RandAR-XL [32]	775M	2.25	314.2	0.80	0.60
		RandAR-XXL [32]	1.4B	2.15	322.0	0.79	0.62
		STAR-B	261M	2.24	295.3	0.82	0.57
		STAR-L	461M	1.98	322.0	0.82	0.58
		STAR-XL	955M	1.65	333.2	0.80	0.62
		STAR-XXL	1.5B	1.55	338.8	0.81	0.62

Fig. 5: Class-conditional generation of STAR-XXL on ImageNet-1k at 256 × 256.

using validation images, in which a random connected set of tokens is dropped at given masking ratios, treating the remaining tokens as partial observations. Further implementation details are described in the supplementary material.

4.2 Main Results

Class-conditional Image Generation In Table 1, we compare STAR with state-of-the-art class-conditional image generation models trained on ImageNet-1k at 256 × 256. We observe that simple modification of STAR, introducing the traversal order of the uniform spanning tree to conventional AR models, such as LlamaGen [40], can achieve performance that is sufficiently comparable with state-of-the-art models without increasing the training complexity or architectural changes, such as the sequence order annealing strategy from random permutation to raster-scan order that requires additional hyperparameter

Table 2: Further comparison of class-conditional generative models [32, 34, 54] on ImageNet-1k at 256×256 with image inpainting. We report the average scores of each metric across masking ratios from 0.1 to 0.9, in 0.1 increments, on the image inpainting.

Model Type	Model	# Params	Generation				Inpainting			
			FID(↓)	IS(↑)	P(↑)	R(↑)	FID(↓)	IS(↑)	P(↑)	R(↑)
Diffusion	DiT-XL/2 [34]	675M	2.27	278.2	0.83	0.57	4.58	50.4	0.74	0.63
Raster-scan AR	RAR-L [54]	461M	1.70	299.5	0.81	0.60	3.56	86.7	0.79	0.60
	RAR-XL [54]	955M	1.50	306.9	0.80	0.62	3.40	88.2	0.79	0.61
Randomized AR	RandAR-L [32]	343M	2.55	288.8	0.81	0.58	2.66	58.8	0.78	0.61
	RandAR-XL [32]	775M	2.25	314.2	0.80	0.60	2.58	60.3	0.77	0.62
	STAR-L	461M	1.98	322.0	0.82	0.58	2.20	110.1	0.82	0.58
	STAR-XL	955M	1.65	333.2	0.80	0.62	2.07	111.3	0.81	0.60

Fig. 6: Qualitative image inpainting examples of class-conditional generative models.

search like RAR [54] and the block-wise causal attention mask and tailored tokenizer like VAR [44]. In addition, STAR outperforms RandAR [32], which adopts random permutation of sequence orders during training, demonstrating the effectiveness of spanning trees over random permutation in sequence order randomization. Especially, given that MAR [26] also adopts random permutation, the notable performance of STAR trained with cross-entropy loss and a discrete VQ tokenizer suggests a promising future extension with a diffusion loss and a continuous VAE tokenizer. Figure 5 illustrates sampled results from STAR-XXL, and more examples are included in the supplementary material.

Comparison with the Image Inpainting In Table 2, we further compare the image inpainting performance of STAR with DiT, RAR, and RandAR, as well as their image generation performance. We provide the partial observation as a prefix for STAR and RandAR, and through teacher forcing for RAR, and latent blending in Blended Latent Diffusion [1] for DiT. We observe that RAR and DiT exhibit a significant decrease in inpainting performance relative to their generation performance, and that some samples show low coherence with partial observations as depicted in Figure 6. This demonstrates that exposing bidirectional dependency between image patches only during training, or only with noisy partial observations of the reverse-time diffusion process, is insufficient to achieve native image inpainting. Meanwhile, RandAR maintains the inpainting performance but shows inferior performance metrics compared to STAR in both generation and inpainting. Moreover, we find that scaling the parameter size significantly improves FID at 0.4-0.6 masking ratios, as shown in Figure 7. We speculate that sufficient sample diversity, as reflected in the trend of Recall in

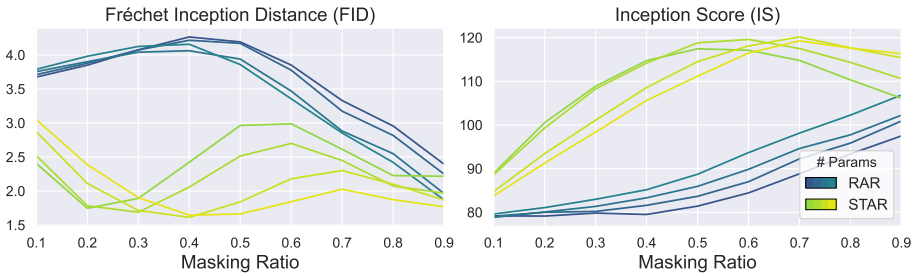


Fig. 7: Detailed comparison of RAR [54] and STAR in the image inpainting across four parameter configurations and varying masking ratios, measured by FID and IS.

Table 1, enables the incorporation of ambiguous yet informative partial observations. Still, RAR consistently performs worse than STAR, and its performance can be restored only at high masking ratios that resemble generation scenarios.

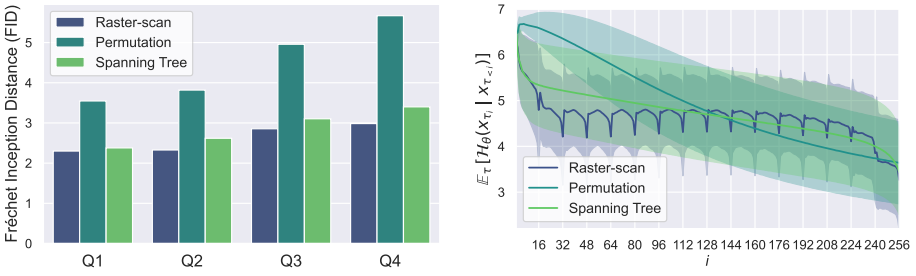
4.3 Ablation Studies and Analysis

Advantage from the Locality STAR utilizes the traversal orders of uniform spanning trees that form a subset of random permutations (i.e $\tau_s\big(\mathcal{T}(G,r)\big) \subset S_N$). Given that RAR utilizes random permutation as a pretext task for the raster-scan order by annealing strategy, $S_N \setminus \tau_s\big(\mathcal{T}(G,r)\big)$ containing sequence orders that do not preserve adjacent region predicting might be similarly beneficial for sequence orders in $\tau_s\big(\mathcal{T}(G,r)\big)$. In this sense, we compare performance changes across sequence order randomization strategies under the same computational budget as STAR-B, as shown in Table 3. However, we find that random permutation cannot outperform the default strategy of STAR, whether using the traversal order of uniform spanning trees at inference time or the annealing strategy. We suspect that sequence orders that do not preserve locality are not advantageous for convergence speed, and the lack of improvement from adopting uniform spanning trees at inference time or after the order annealing indicates that they do not serve as a meaningful pretext task beyond a single predefined sequence order. Therefore, it demonstrates the efficiency of STAR in providing a structured random family of locality-preserving sequence orders that are sparse in random permutation, without requiring a complex annealing strategy.

Advantage from the Center Bias We select random corners as the root node for BFS traversal in STAR from the assumption that starting from non-salient regions is advantageous under center bias, according to the observation in Section 3.2 and prior works [12,13,54] that consistently report that the raster-scan order, which starts from the corner, outperforms other choice of structured sequence orders. We further analyze this design choice by dividing ImageNet classes into four quartiles based on the strength of the center bias, measured by the standard deviation of relative object distances from the image center. Specifically, we measure FID using 50k generated images and reference images from the entire training dataset, only for the subclasses within each quartile, as

Table 3: Change in generation and inpainting performance according to the sequence order randomization strategy. "Start" and "End" denote the sequence order at the start or end, and perform order annealing at 0.5 to 0.75 of the total training steps [54].

Training		Inference	Generation				Inpainting			
Start	End		FID(↓)	IS(↑)	P(↑)	R(↑)	FID(↓)	IS(↑)	P(↑)	R(↑)
Raster-scan	Raster-scan	Raster-scan	2.04	266.5	0.80	0.59	3.91	82.7	0.75	0.60
Permutation	Permutation	Permutation	3.57	291.4	0.78	0.57	2.57	108.7	0.82	0.56
Permutation	Permutation	Spanning Tree	3.58	234.9	0.76	0.60	2.43	105.8	0.83	0.56
Permutation	Spanning Tree	Spanning Tree	2.31	285.2	0.81	0.57	2.35	107.9	0.83	0.56
Spanning Tree	Spanning Tree	Spanning Tree	2.24	295.3	0.82	0.57	2.39	108.8	0.83	0.56



(a) **(b)**

Fig. 8: (a) FID measured in subclasses divided according to the concentration of objects in the central region. Moving from Q1 to Q4, objects in subclasses tend to disperse farther from the center, implying a weaker center bias. (b) Conditional model entropy according to the token position in sequence orders. Sequence orders with better performance tend to have a uniform scale of conditional model entropy across token positions.

shown in Figure 8a. We observe that BFS traversal of uniform spanning trees achieves performance comparable to raster-scan order at higher center bias and consistently outperforms random permutation at all levels. We also find that performance degrades as center bias decreases regardless of sequence order choice, which is likely due to increased uncertainty about object locations resulted from reduced center bias in images. Therefore, these results reveal that randomly selected corners as the root are well-suited to the case with strong center bias, and still generalize better than random permutation under weaker center bias.

Difference in the Conditional Model Entropy We have verified that prior knowledge observed in the image is still preserved in the token representations in Section 3.2, but it does not fully capture differences in the characteristics of sequence orders. In this sense, we compare the conditional model entropy of models trained with raster-scan order, random permutation, and BFS traversal of the uniform spanning tree, based on the token position in the linearized sequence. In Figure 8b, we observe that the conditional model entropy of the AR model trained on the BFS traversal of the uniform spanning tree exhibits characteristics similar to those of the AR model with raster-scan order based on the scale change with token position. Specifically, random permutation shows high conditional model entropy at the initial token positions, then decays quickly after half of the

Table 4: Comparison of the DFS and BFS with root selection strategies in the rejection sampling for postfix completion, measured by the number of trials and failure rate.

Metric	Traversal	Root	Masking Ratio								
			0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
Trial (#)	DFS	Random	60.45	52.09	44.63	36.87	29.31	23.28	13.62	6.12	2.22
		Farthest	89.37	80.25	70.45	53.70	23.33	19.70	14.29	6.19	2.23
	BFS	Random	5.59	3.56	2.76	2.11	1.60	1.49	1.44	1.28	1.15
		Farthest	4.41	3.03	2.50	1.95	1.57	1.51	1.43	1.27	1.15
Failure (%)	DFS	Random	50.1	41.8	33.0	24.9	17.4	11.6	3.0	0.0	0.0
		Farthest	82.3	70.0	57.9	40.4	12.3	9.8	3.6	0.0	0.0
	BFS	Random	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		Farthest	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

token positions, unlike others, which show relatively constant scale across token positions. Given that both cases show better performance compared to random permutation, we speculate that uniformity of the prediction difficulty yields a beneficial improvement in convergence speed, similar to the claim of AIM [12].

Overhead from the Spanning Tree STAR incurs additional overhead from sampling a uniform spanning tree and performing BFS on each instance, both in training and in inference. Still, we find that 0.56 ms is incurred per instance on average, corresponding to $< 0.0004\%$ of the total inference time across four STAR configurations, which is sufficiently small to be negligible. However, rejection sampling for postfix completion can induce multiple trials of uniform spanning tree sampling, so we check the efficiency of rejection sampling depending on the choice of DFS and BFS with the farthest root selection strategy, as shown in Table 4. We measure the number of trials required to sample a spanning tree that satisfies the condition for postfix completion under the corresponding traversal order, for 50k random connected masks with masking ratios ranging from 0.1 to 0.9 in increments of 0.1. The maximum number of trials for each random mask is bounded at 100, and cases that exceed this bound are considered failures. We find that DFS requires significantly more trials than BFS and fails more frequently, demonstrating the validity of selecting BFS as the default traversal strategy in STAR. In particular, we find that we can sample the spanning tree with fewer than 5 trials using BFS with the farthest root selection strategy, which reduces the number of trials at relatively small masking ratios.

5 Discussion

We investigate STAR primarily with a focus on visual generation, but AR modeling also has another research direction: representation learning [7, 12]. In this context, we train models with the same configuration as iGPT-S [7] using only the CIFAR-100 dataset [23] for 100k steps, while changing the sequence order. In Figure 9, we measure the top-5 test accuracy from the linear probe using mean-pooled intermediate hidden states at the 2/3 point, along with test-set

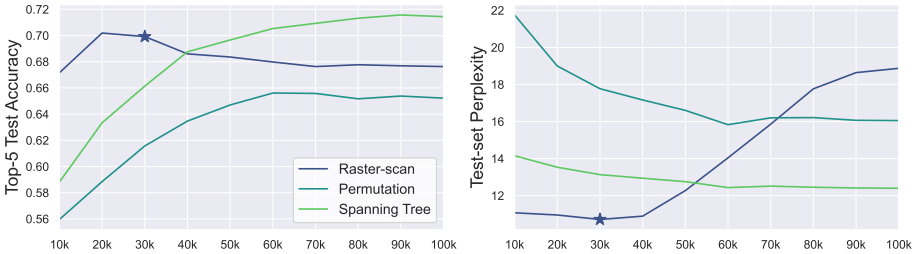


Fig. 9: Comparison of sequence orders in representation learning. We mark the point where validation loss starts to increase, which is observed only for the raster-scan order.

perplexity, for checkpoints saved at 10k step intervals. We confirm that uniform spanning trees achieve superior linear probe results while maintaining robustness against overfitting compared to raster-scan order, and achieve reasonable convergence speed compared to random permutation. This suggests that future work on STAR can be investigated within the context of representation learning.

6 Conclusion

In this work, we propose Spanning Tree Autoregressive (STAR) modeling, which adopts the BFS traversal order of the uniform spanning tree rooted at the corner of the image patch lattice as the sequence order randomization strategy for AR visual generation models. It structurally reflects prior knowledge of images, thereby maintaining generation performance without a complex training strategy, while still supporting flexible sequence orders to achieve region-level modification, which were the weaknesses and strengths of permutation AR modeling. Notably, a remarkable feature of STAR is its simplicity, which requires only minimal changes to the widely adopted conventional AR modeling architecture to support both robust image generation and image inpainting. We hope this simplicity of STAR can serve as a basic building block for future AR visual modeling work and be extended to early-fusion multimodal generative models.

Acknowledgements

This work was supported by LG AI Research. This work was supported by the Korea Institute of Science and Technology (KIST) Institutional Program (Project No. 26E0061). This work was partly supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean Government (MSIT) (No. RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)), the Technology Innovation Program (RS-2025-25456760, Development of a humanoid robot specialized in chemical processes based on AI foundation model) funded By the Ministry of Trade, Industry and Resources (MOTIR, Korea). We express special thanks to Korea Association for AI & ICT Promotion (KAIT) GPU project. The ICT at Seoul National University provides research facilities for this study.

References

1. Avrahami, O., Fried, O., Lischinski, D.: Blended latent diffusion. ACM transactions on graphics (TOG) **42**(4), 1–11 (2023)
2. Bao, F., Nie, S., Xue, K., Cao, Y., Li, C., Su, H., Zhu, J.: All are worth words: A vit backbone for diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 22669–22679 (2023)
3. Besnier, V., Chen, M., Hurych, D., Valle, E., Cord, M.: Halton scheduler for masked generative image transformer. arXiv preprint arXiv:2503.17076 (2025)
4. Borji, A., Tanner, J.: Reconciling saliency and object center-bias hypotheses in explaining free-viewing fixations. IEEE transactions on neural networks and learning systems **27**(6), 1214–1226 (2015)
5. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
6. Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T.: Maskgit: Masked generative image transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11315–11325 (2022)
7. Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I.: Generative pretraining from pixels. In: International conference on machine learning. pp. 1691–1703. PMLR (2020)
8. Dehghani, M., Djolonga, J., Mustafa, B., Padlewski, P., Heek, J., Gilmer, J., Steiner, A.P., Caron, M., Geirhos, R., Alabdulmohsin, I., et al.: Scaling vision transformers to 22 billion parameters. In: International conference on machine learning. pp. 7480–7512. PMLR (2023)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
10. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021)
11. Dosovitskiy, A.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
12. El-Nouby, A., Klein, M., Zhai, S., Bautista, M.A., Shankar, V., Toshev, A., Susskind, J.M., Joulin, A.: Scalable pre-training of large autoregressive image models. In: Proceedings of the 41st International Conference on Machine Learning. pp. 12371–12384 (2024)
13. 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)
14. Gao, S., Zhou, P., Cheng, M.M., Yan, S.: Masked diffusion transformer is a strong image synthesizer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 23164–23173 (2023)
15. Henighan, T., Kaplan, J., Katz, M., Chen, M., Hesse, C., Jackson, J., Jun, H., Brown, T.B., Dhariwal, P., Gray, S., et al.: Scaling laws for autoregressive generative modeling. arXiv preprint arXiv:2010.14701 (2020)
16. Heo, B., Park, S., Han, D., Yun, S.: Rotary position embedding for vision transformer. In: European Conference on Computer Vision. pp. 289–305. Springer (2024)
17. 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** (2017)

18. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
19. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
20. Huang, J., Mumford, D.: Statistics of natural images and models. In: Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149). vol. 1, pp. 541–547. IEEE (1999)
21. Kaplan, J., McCandlish, S., Henighan, T., Brown, T.B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., Amodei, D.: Scaling laws for neural language models. arXiv preprint arXiv:2001.08361 (2020)
22. Kilian, M., Jampani, V., Zettlemoyer, L.: Computational tradeoffs in image synthesis: Diffusion, masked-token, and next-token prediction. arXiv preprint arXiv:2405.13218 (2024)
23. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
24. Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., Aila, T.: Improved precision and recall metric for assessing generative models. Advances in neural information processing systems **32** (2019)
25. Lee, D., Kim, C., Kim, S., Cho, M., Han, W.S.: Autoregressive image generation using residual quantization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11523–11532 (2022)
26. Li, H., Yang, J., Li, G., Wang, H.: Autoregressive image generation with randomized parallel decoding. arXiv preprint arXiv:2503.10568 (2025)
27. 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)
28. Liu, X., Hao, S., Qi, X., Hu, T., Wang, J., Xiao, R., Yao, Y.: Elucidating the design space of language models for image generation. arXiv preprint arXiv:2410.16257 (2024)
29. Liu, Y., Qu, L., Zhang, H., Wang, X., Jiang, Y., Gao, Y., Ye, H., Li, X., Wang, S., Du, D.K., et al.: Detailflow: 1d coarse-to-fine autoregressive image generation via next-detail prediction. arXiv preprint arXiv:2505.21473 (2025)
30. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
31. Ou, J., Nie, S., Xue, K., Zhu, F., Sun, J., Li, Z., Li, C.: Your absorbing discrete diffusion secretly models the conditional distributions of clean data. arXiv preprint arXiv:2406.03736 (2024)
32. Pang, Z., Zhang, T., Luan, F., Man, Y., Tan, H., Zhang, K., Freeman, W.T., Wang, Y.X.: Randar: Decoder-only autoregressive visual generation in random orders. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 45–55 (2025)
33. Pannatier, A., Courdier, E., Fleuret, F.: σ-gpts: A new approach to autoregressive models. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 143–159. Springer (2024)
34. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)
35. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
36. 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)

37. Roheda, S., Chowdhury, R., Bala, A., Jaiswal, R.: Cart: Compositional autoregressive transformer for image generation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 40, pp. 8740–8750 (2026)
38. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. Advances in neural information processing systems **29** (2016)
39. Shrock, R., Wu, F.Y.: Spanningtrees on graphs and lattices in d dimensions. Journal of Physics A: Mathematical and General **33**(21), 3881 (2000)
40. 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)
41. Szabó, G., Horváth, A.: Mitigating the bias of centered objects in common datasets. In: 2022 26th International Conference on Pattern Recognition (ICPR). pp. 4786–4792. IEEE (2022)
42. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
43. Team, C.: Chameleon: Mixed-modal early-fusion foundation models. arXiv preprint arXiv:2405.09818 (2024)
44. 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)
45. 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)
46. Uria, B., Côté, M.A., Gregor, K., Murray, I., Larochelle, H.: Neural autoregressive distribution estimation. Journal of Machine Learning Research **17**(205), 1–37 (2016)
47. Van Den Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: International conference on machine learning. pp. 1747–1756. PMLR (2016)
48. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. Advances in neural information processing systems **30** (2017)
49. Wang, X., Dufour, N., Andreou, N., Cani, M.P., Abrevaya, V.F., Picard, D., Kalogeiton, V.: Analysis of classifier-free guidance weight schedulers. Transactions on Machine Learning Research Journal (2024)
50. Wilson, D.B.: Generating random spanning trees more quickly than the cover time. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. pp. 296–303 (1996)
51. Xue, S., Xie, T., Hu, T., Feng, Z., Sun, J., Kawaguchi, K., Li, Z., Ma, Z.M.: Any-order gpt as masked diffusion model: Decoupling formulation and architecture. arXiv preprint arXiv:2506.19935 (2025)
52. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems **32** (2019)
53. Yu, J., Li, X., Koh, J.Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., Wu, Y.: Vector-quantized image modeling with improved vqgan. arXiv preprint arXiv:2110.04627 (2021)
54. Yu, Q., He, J., Deng, X., Shen, X., Chen, L.C.: Randomized autoregressive visual generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18431–18441 (2025)