Towards Better & Faster Autoregressive Image Generation: From the Perspective of Entropy

Xiaoxiao Ma[1] **Feng Zhao**[1†] **Pengyang Ling**[1] **Haibo Qiu**[2*] **Zhixiang Wei**[1]
Hu Yu[1] **Jie Huang**[1] **Zhixiong Zeng**[2] **Lin Ma**[2]
[1]University of Science and Technology of China [2]Meituan

Figure 1: Top row: Our method generates images with finer details and better structure. Bottom row: Combined with existing acceleration methods, ours reduces inference cost by 15%. (Left two pairs are from LlamaGen [1]; right from Lumina-mGPT [2]. Inference steps and latency are reported.)

Abstract

In this work, we first revisit the sampling issues in current autoregressive (AR) image generation models and identify that image tokens, unlike text tokens, exhibit lower information density and non-uniform spatial distribution. Accordingly, we present an entropy-informed decoding strategy that facilitates higher autoregressive generation quality with faster synthesis speed. Specifically, the proposed method introduces two main innovations: 1) dynamic temperature control guided by spatial entropy of token distributions, enhancing the balance between content diversity, alignment accuracy, and structural coherence in both mask-based and scale-wise models, without extra computational overhead, and 2) entropy-aware acceptance rules in speculative decoding, achieving near-lossless generation at about 85% of the inference cost of conventional acceleration methods. Extensive experiments across multiple benchmarks using diverse AR image generation models demonstrate the effectiveness and generalizability of our approach in enhancing both generation quality and sampling speed. Code is available at https://github.com/krennic999/ARsample.

1 Introduction

Autoregressive (AR) modeling, as the mainstream in language generation [3, 4, 5, 6], has recently demonstrate strong potential in visual generation [2, 1], offering improved scalability [7] and potential for unified vision-language modeling [8, 9]. In this paradigm, images are first quantized into discrete

*Project lead; † Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

token sequences [10, 11], which are then generated either token-by-token in a raster-scan order [8, 12], or in parallel through multi-token generation strategies [7, 13, 14].

Unlike diffusion or flow models that regress continuous tokens [15, 16, 17], AR models learn the probabilistic distribution over discrete vocabulary, and sampling strategies (e.g., top-K, top-p [18]) are required to specify a token, which is essential and significantly impacts the quality and characteristics of generated content. Within the area of language modeling, strategies have been proposed to augment reasoning capabilities and mitigate hallucinations, including logit shaping [19, 20], contrastive decoding [21, 22], leveraging model-specific features [23], and search-based methods [24, 25], which emphasize answer accuracy over generation diversity.

However, a clear distinction exists between image and language: *images exhibit lower information density and highly non-uniform spatial information distribution*, as shown in Fig. 2 (a), making language-oriented methods suboptimal for image generation. This mismatch often leads to a trade-off between diversity in image contents and text consistency. As observed in [2, 26], increasing randomness (*e.g.,* high top-K) helps enrich visual content but compromises structural stability, leading to artifacts, distorted structures, or chaotic textures. Conversely, reducing randomness stabilizes structure and improves alignment, but often yields flat, oversmoothed details, or simplistic background. How to balance randomness and determinism during sampling is thus critical for high-quality image generation. Unfortunately, existing methods typically rely on uniform sampling approaches like fixed top-K or top-p, overlooking the inherent spatial information imbalance, which limits their ability to achieve high-quality images.

In this work, we aim to leverage the uneven distribution of information in images, and propose a *sampling method specifically for autoregressive image generation*. We observe that entropy of predicted logits effectively reflects information density in image during generation—low entropy corresponds to large homogeneous regions, while high entropy highlights content-rich areas such as foreground objects and complex backgrounds (see Fig. 3). An intuitive idea is to encourage higher randomness in low-entropy regions, while applying stricter sampling in high-entropy areas. This helps balance image richness with structural stability, and also allocating fewer inference resources to low-entropy regions, which enables further acceleration with minimal impact on generation quality. Unlike [27], which applies entropy to control sampling randomness in super-resolution to modulate stochasticity, our work leverages entropy to guide autoregressive generation dynamics.

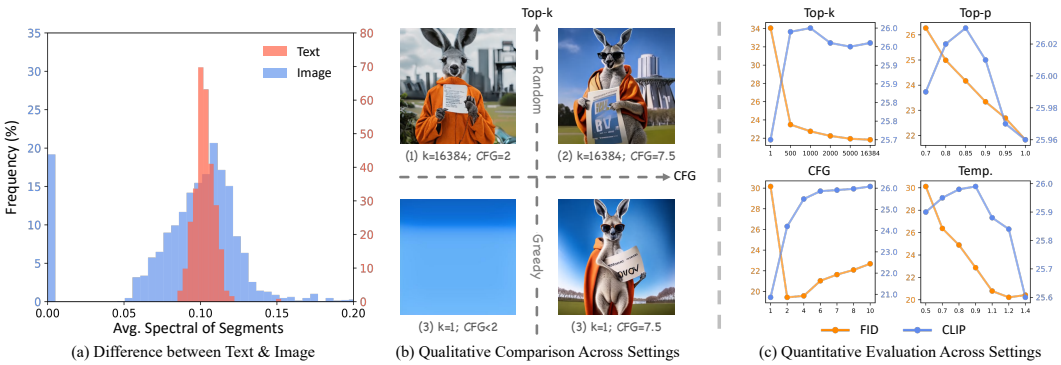
Building on this observation, we propose an entropy-aware sampling strategy that adjusts token distributions dynamically during inference. By computing the entropy of each predicted token distribution, we assign adaptive temperatures—injecting more randomness in low-entropy (simple) regions and applying stricter sampling in high-entropy (complex) areas. This improves the balance between image quality, structural stability, and text-image alignment without additional training or inference cost. Moreover, our method generalizes well to a variety of autoregressive frameworks based on discrete token prediction, including mask-based and scale-wise generation. We also extend the entropy-aware idea to acceleration: by incorporating entropy-dependent acceptance in speculative decoding, we reduce inference cost to 85% of standard baselines with minimal quality loss. We summarize our contributions as follows:

1. Motivated by the observation that image information is sparse and unevenly distributed, which can be reflected by the entropy of tokens, we introduce an entropy-driven sampling strategy tailored for AR image generation that dynamically adapts sampling behavior based on entropy.
2. In contrast to conventional sampling methods like top-K or top-p, our approach enhances image quality and structural stability without modifying the model or increasing inference cost, and benefits multiple types of AR generation frameworks.
3. We further extend the entropy-aware perspective to speculative decoding, achieving a 15% reduction in inference time while maintaining visual fidelity across multiple benchmarks.

2 Related works

2.1 Autoregressive image generation

Early work [28] generates images directly at pixel level. Later approaches adopt a two-stage pipeline: images are first quantized into discrete tokens [10, 11], then generated with Transformers in raster



(a) Difference between Text & Image (b) Qualitative Comparison Across Settings (c) Quantitative Evaluation Across Settings

Figure 2: (a) Comparison of information density between image and text. Histogram of average frequency-domain embeddings from LlamaGen [1] (image) and Qwen2 [6] (text) show the uneven spatial distribution in images with a large amount of low-frequency components. (b) Qualitative results under various configurations. High CFG (Classifier-Free Guidance) or low top-K often harms fidelity, while lower CFG with higher top-K improves fidelity but may reduce text-image consistency. (c) Quantitative evaluation of LlamaGen under different sampling settings.

order [29, 30, 31, 32, 33, 34]. Recent efforts scale this paradigm with larger models and stronger conditioning. LlamaGen [1] provides class and text-conditioned baselines; Lumina-mGPT [2] and Anole [35] fine-tune Chameleon [12] for improved text-conditioned generation. Unified frameworks further bridge understanding and generation [8, 9, 36, 37] in a single Transformer. Meanwhile, image tokenizers have evolved for better reconstruction [38, 39, 40, 41] or multimodal integration [42, 43].

While proven effective, the vanilla autoregressive paradigm suffers from slow and rigid next-token prediction. To improve efficiency, recent studies explore more strategies, including multi-token prediction via random masking [14, 44, 45, 46], coarse-to-fine modeling [7, 13, 47, 48, 49]or hybrid approaches [50, 51]. Nonetheless, vector-quantized models still rely on sampling from token distributions, making generation quality sensitive to the sampling strategy.

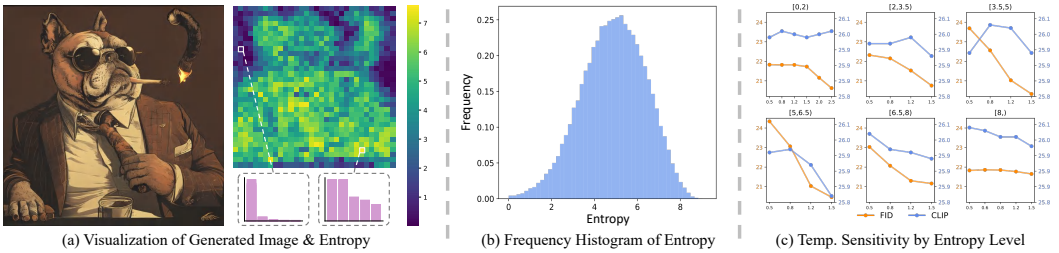
2.2 Sampling strategies in autoregressive models

Transformers model the probability distribution over tokens, requiring specific sampling strategies to obtain concrete outputs. Common approaches in language modeling include top-k [52] and top-p [18] sampling, which truncate the candidate space by rank or cumulative probability. EDT [20] dynamically adjusts temperature based on entropy to balance diversity and precision. Other approaches explore repetition penalties [53], contrastive decoding [22], speculative decoding [54, 55], and search-based techniques [56, 25, 57, 58] to reduce hallucination or speed up inference.

In visual generation, a higher degree of randomness is often needed to produce more realistic and detailed content. LlamaGen [1] and Lumina-mGPT [2] demonstrate that much larger top-k values than those used in language models help avoid over-smoothed and low-detail outputs. Recent methods [26, 59] apply speculative [54] or parallel decoding [60, 61] to accelerate image synthesis. PURE [27] designs a top-k strategy based on token entropy and detail levels to improve autoregressive super-resolution. However, they overlook the highly uneven spatial information distribution in images during generation, and do not tailor decoding for autoregressive image generation.

3 Methods

In this section, we first introduce basic of autoregressive image generation in Sec. 3.1. Starting from the difference between image and text generation, we present our method from an entropy-based perspective by adjusting token-level randomness during generation (Sec. 3.2, Sec. 3.3), and further extend this view to acceleration (Sec. 3.4).



(a) Visualization of Generated Image & Entropy (b) Frequency Histogram of Entropy (c) Temp. Sensitivity by Entropy Level

Figure 3: (a) Entropy map during generation: complex regions exhibit higher entropy (more dispersed probabilities), while simpler areas show lower entropy. (b) Histogram of entropy distribution on COCO val2017 (from LlamaGen Stage II). (c) Varying temperature by entropy range affects FID and CLIP score: lower-entropy tokens benefit from higher temperatures, and vice versa.

3.1 Preliminaries and motivation

Autoregressive image generation. In a typical autoregressive generation process, an image $I \in \mathbb{R}^{H\times W\times 3}$ is quantized into a set of discrete tokens $(x_1, x_2, ..., x_{h\times w})$, where each token $x_i \in [V]$, V denotes the size of the VQ-VAE codebook. The image tokens are generated sequentially by a transformer, with the $i+1$-th token x_{i+1} conditioned on the previously generated tokens. This process is modeled as $\prod_{i=1}^{hw-1} p(x_{i+1} \mid x_{1:i})$, where $x_{1:i} = (x_1, x_2, ..., x_i)$, and $p(x \mid x_{1:i})$ represents a categorical distribution over token at position $i+1$. In text-conditioned generation tasks, the full image sequence is generated conditioned on a prefix of text tokens. At each step, a sampling method such as top-K or top-p is applied to select a token from $p(x \mid x_{1:i})$. And choice of sampling strategy can significantly affect the quality of generated image, as illustrated in Fig. 2 (b).

Difference between image & text generation. Compared to text, *the information density in images is lower and highly non-uniform*. For example, images often contain large regions of solid color or visually similar content, while nearby tokens in text are typically distinct. To illustrate the difference, we segment the embedding sequences of both images and texts into equal lengths, compute the average frequency spectrum per segment, and visualize the distributions in Fig. 2 (a). The average frequency spectrum distribution of image segments is more dispersed, with a large amount of low-frequency areas; whereas textual segments demonstrate more compact and uniform distributions.

This discrepancy poses a challenge for sampling: fixed parameters like top-K or top-p, though effective in language generation, perform suboptimally when simply applying them across all image tokens. They fail to account for spatial variability, resulting in regional artifacts: overly deterministic sampling may lose fine details, producing flat regions, excessive smoothness, or simplistic backgrounds; while excessively random sampling compromises semantic consistency and structural coherence, causing artifacts, distorted limbs, or chaotic textures.

Relationship between entropy & image contents. We demonstrate that the *entropy of predicted token distribution serves as an effective indicator of local information density* in an image. Specifically, we compute the entropy ϵ of log-likelihood over all V codebook entries at each generation step as:

$$\epsilon = -\sum_{k=1}^{V} p_k \log(p_k). \tag{1}$$

As shown in Fig. 3, regions with simple content (*e.g.,* solid colors) typically exhibit lower entropy, while more complex foreground (*e.g.,* objects, structures, and textures) areas have higher entropy. Low-entropy regions correspond to peaked distributions over a few tokens, indicating high model confidence. Conversely, high-entropy regions display more uniform distribution, reflecting greater uncertainty in token selection and higher information density. These observations validate entropy as a reliable proxy for measuring information density in images.

3.2 Entropy-aware dynamic temperature

Building on the observation in Sec. 3.1, we further investigate how regions with different entropy levels affect image quality and the optimal sampling strategy. Under a simple experimental setup, we adopt [1] to analyze the entropy distribution of logits during generation, discretize it into intervals,

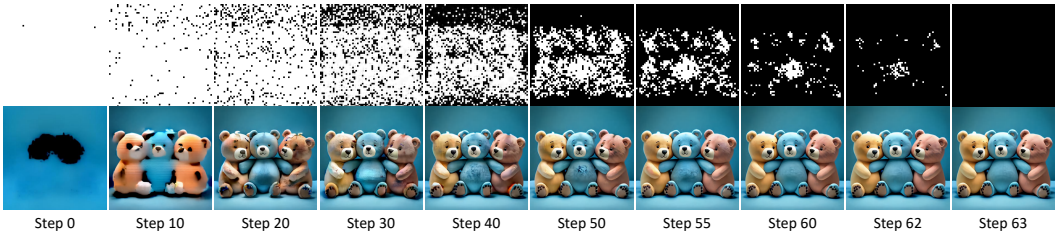


Figure 4: During generation of mask-based model [44], a large number of early steps (0∼50) are allocated to compute tokens in simple regions, while only a few later steps (e.g., 50∼63) for generating complex content. This often leads to degraded quality in the main visual subjects.

and adjust token temperature within each interval to control sampling randomness. We then examine the relationship between image quality and text alignment via FID and CLIP-Score as indicators, results can be seen in Fig. 3 (c). Our findings are as follows:

1. Across most entropy intervals, adjusting randomness leads to a trade-off between image quality and text alignment, especially for tokens in the entropy range of [2,8].
2. In high-entropy regions (>5), lower randomness helps improve text-image consistency.
3. In regions with extreme low entropy (<2), increasing sampling randomness consistently improves visual quality, while having negligible impact on text alignment.

These findings suggest that token-level sampling should be entropy-aware: during inference, tokens with *lower entropy* should be assigned relatively *higher randomness* to enhance the quality and visual richness of generated image, while *high-entropy* tokens should be sampled *more cautiously* to preserve clear structure, details and text alignment.

To better adapt to real-world inference, we introduce a dynamic temperature mechanism that adjusts sampling randomness on a per-token basis. Specifically, after computing the entropy of predicted distribution at each position, we determine a temperature value with predefined mapping function:

$$T = T_0 e^{-\frac{\epsilon}{\alpha}} + \theta, \tag{2}$$

where ϵ denotes the entropy at current token position, T_0 represents the maximum temperature, θ sets the lower bound, and α controls the decay rate of temperature with increasing entropy. See Sec. 4.4 for further analysis and discussion. Subsequently, the resulting temperature T is then applied to rescale the predicted logits as follows:

$$\tilde{p}_i = \frac{p_i}{T}. \tag{3}$$

Then, by applying $softmax(\tilde{p}_i)$, the differences between logits of different tokens are amplified (when T<1) or reduced (when T>1), which makes the probability distribution more concentrated or spread out, achieving the dynamic adjustment of sampling based on the region's content distribution.

3.3 Adaptation to more AR models

Additionally, many recent methods deviate from strict next-token prediction and instead adopt paradigms such as *mask-prediction* or *scale-wise* generation. We show that the proposed entropy-based strategy remains effective in these settings. After obtaining multiple token-level logit distributions from the transformer, we directly apply Eq. (2) to them. As shown in Table 1, this approach consistently improves performance across standard evaluation metrics. Moreover, for different paradigms, specific designs can be incorporated to further enhance performance:

Mask-prediction models. For mask-based models such as [44, 45], a full probability distribution over all image tokens is obtained at each forward step. After sampling, a dynamic masking mechanism based on token confidence is applied. We observe that this masking strategy also has a significant impact on the quality of the generated results (see Fig. 4). Specifically, a soft categorical distribution is used to select k tokens from all candidates to be accepted at the current timestep t:

$$\text{conf} = \log p_t + T \cdot g, \tag{4}$$

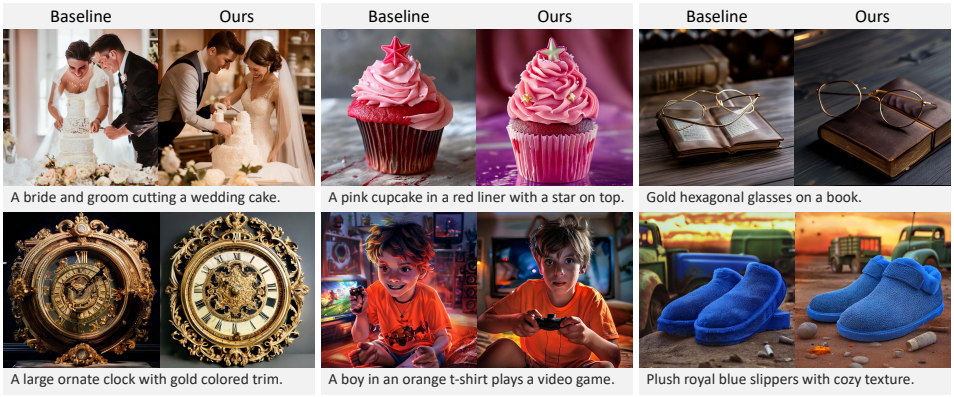


Figure 5: Visual comparison on next-token model. Examples are from Lumina-mGPT, proposed method ("Ours") maintains richer content while offering more accurate structure and finer details.

Table 1: Performance of sampling strategies on various models. "Baseline" is original sampling; "+Prob./Ours" refers to Sec. 3.2; "+Masking/+Scale-wise" are paradigm-specific from Sec 3.3.

Method	Config.	FID↓	CLIP-Score↑	DPG↑	HPSv2.1↑
SDv2.1 [15]	-	22.87	26.31	68.09	26.38
PixArt-α [62]	-	33.23	25.70	71.52	30.04
SDXL [63]	-	23.20	26.46	74.21	28.54
SDv3-medium [17]	-	29.82	26.24	85.85	30.22
LlamaGen [1]	Baseline	21.94	25.95	43.51	21.24
	Ours	**20.36**	**25.96**	**48.63**	**21.39**
Lumina-mGPT [2]	Baseline	29.15	26.04	79.68	**28.92**
	Ours	**27.44**	**26.25**	**79.77**	28.87
Meissonic [44]	Baseline	53.61	25.27	63.83	29.33
	+Prob.	**48.37**	25.49	66.19	29.94
	+Masking.	48.43	**25.54**	**67.08**	**30.04**
STAR [13]	Baseline	35.05	25.43	70.25	28.79
	+Prob.	32.75	25.56	70.83	28.93
	+Scale-wise	**32.37**	**25.61**	**70.86**	**29.06**

where p is the predicted token probability, T is the dynamic temperature defined in Eq. (2), $g \sim \text{Gumbel}(0, 1)$ is sampled from standard Gumbel distribution.

$$\mathcal{M}_t = \text{conf} < \text{TopK}(\text{conf} \odot \tilde{\mathcal{M}}_{t-1}, k). \tag{5}$$

Here, $\text{TopK}(\text{conf}, k)$ returns the k-th highest confidence score, and tokens with lower confidence are masked out. $\mathcal{M}_t$ and $\mathcal{M}_{t-1}$ represent the accepted token masks at the current and previous timesteps, respectively, while $\tilde{\mathcal{M}}_{t-1}$ denotes the element-wise negation of $\mathcal{M}_{t-1}$. The number of accepted tokens k at each timestep t is determined by a predefined scheduler. This design further encourages randomness in low-entropy regions while enhancing accuracy in high-entropy regions, leading to improved image quality.

Scale-wise models [7, 13, 47, 48] generate tokens within each scale simultaneously. We find that assigning greater randomness to earlier scales while reducing randomness at later scales yields more accurate results without compromising image richness. Specifically, we define a temperature term that decreases as the scale increases. For the tokens at s-th scale, T_s is calculated as:

$$T_s = T \cdot [1 - \beta \cdot (s - \lfloor S/2 \rfloor)], \tag{6a}$$

$$p_s = \frac{p_s}{T_s}, \tag{6b}$$

where s denotes the scale index and $s \in \{1, 2, \ldots, S\}$; p_s is the logits of tokens at s-th scale, with shape of $h_s \times w_s \times V$. T is dynamic temperature defined in Eq. (2). β controls the decay rate of T_s across scales and is set to 0.3 in experiments.

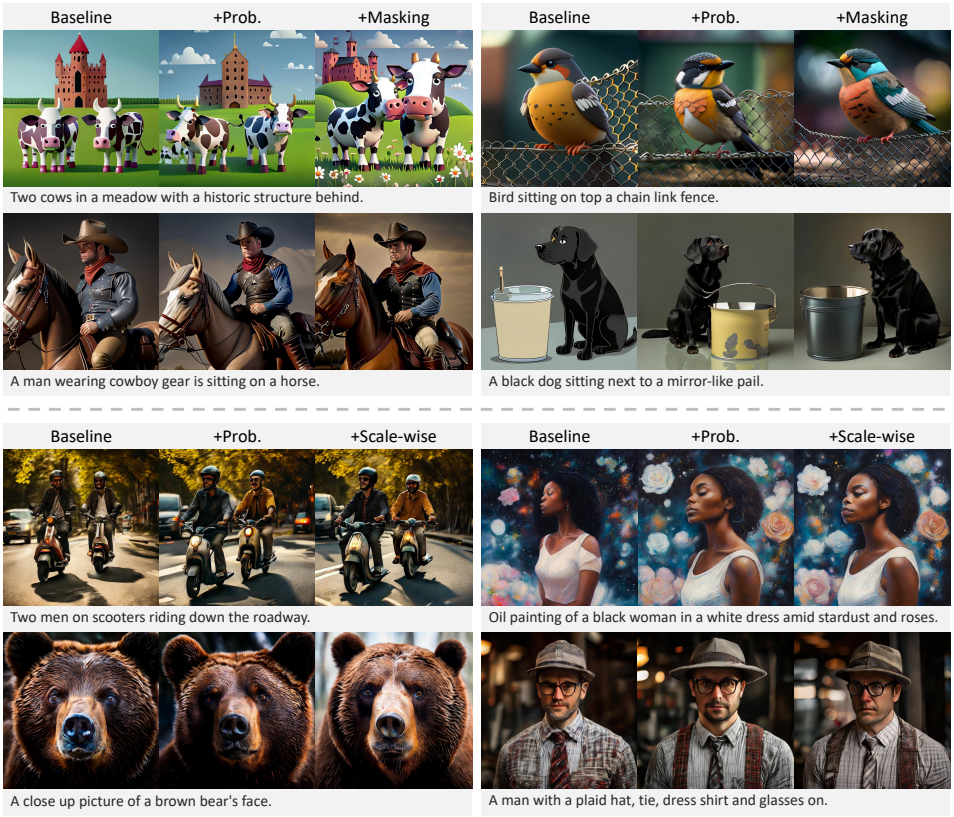


Figure 6: Visual comparison on mask-based (top) and scale-based model (bottom) from Meissonic and STAR. Proposed method provides better visual quality in structure and detail.

3.4 Autoregressive acceleration

We further explore the use of entropy to accelerate autoregressive generation. Existing speculative decoding approaches [26, 59] typically generate multiple candidate tokens via a draft model, followed by a verification step using a target model. When the draft and target share the same model, the process reduces to comparing the confidence scores from two consecutive iterations. Specifically, the probability at the $(j-1)$-th step, $p(x \mid x_{1:i-1}^{(j-1)})$, and the j-th step, $p(x \mid x_{1:i-1}^{(j)})$, are compared. The acceptance probability of the token x_i is then computed based on these two distributions:

$$p = \min\left(1, \frac{p_\theta(x_i^{(j)} \mid x_{1:i-1}^{(j)})}{p_\theta(x_i^{(j-1)} \mid x_{1:i-1}^{(j-1)})}\right). \tag{7}$$

In practice, a token is accepted if $p > r$, where r is drawn from a uniform distribution $\mathcal{U}[0, 1]$, which naturally balances randomness required for sampling diversity and accuracy during generation.

To make the process entropy-aware, we propose a simple modification to the acceptance rule. Since low-entropy regions are more predictable and allow higher randomness, while high-entropy regions require stricter verification, we scale the threshold r by an entropy-based factor ϵ/e, where e is constant. This dynamic adjustment enables more efficient generation based on local uncertainty.

To improve stability, we rewrite r as $0.5 + (r - 0.5)$, treating it as $0.5 + \mathcal{U}[-0.5, 0.5]$, where the noise term controls acceptance randomness. We scale this term with a decaying factor $(1 - \lambda \cdot \epsilon)$, where ϵ is the entropy, making high-entropy tokens more deterministically verified, while low-entropy areas retain near-uniform. Combining both strategies above, the final acceptance rule can be formulated as:

$$p > \frac{\epsilon}{e}\left[0.5 + (r - 0.5)(1 - \lambda \cdot \epsilon)\right], \tag{8}$$

where ϵ is entropy at the current token, e and λ are constants set to 8 and 16, respectively.

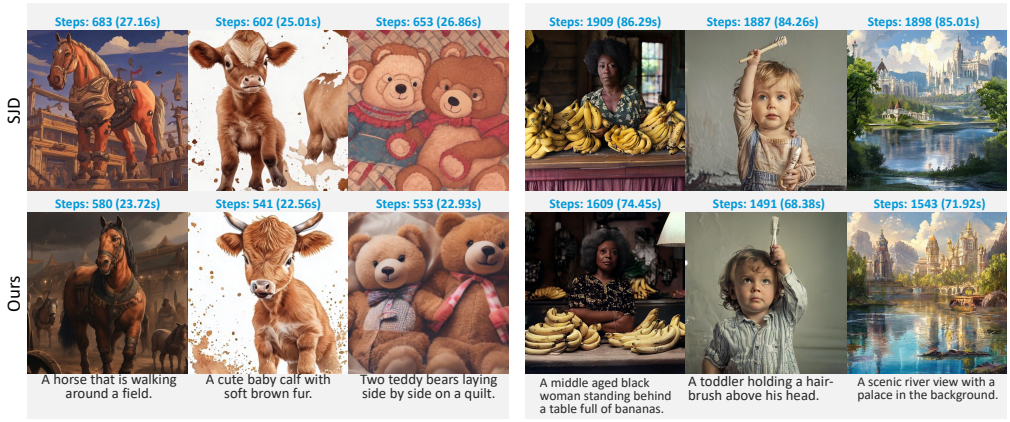


Figure 7: Results on acceleration. We report the inference steps ("Steps") and latency. Our method achieves similar image quality while using only 85% of the baseline's ("SJD") inference cost.

Table 2: Quantitative comparison of acceleration on COCO17-val. "Vanilla" refers no acceleration.

Model	Config.	Avg. Latency [s]↓	Avg. Steps↓	FID↓	CLIP-Score↑
LlamaGen [1]	Vanilla	44.52	1024	53.42	21.47
	SJD [26]	26.52	626.9	54.49	21.45
	Ours	**22.04**	**535.5**	54.60	21.51
Lumina-mGPT [2]	Vanilla	169.72	4165	29.15	26.04
	SJD [26]	84.97	1854.5	30.76	26.09
	Ours	**72.35**	**1594.5**	30.89	26.10

This dynamic acceptance criterion allocates inference budget more efficiently—being more permissive in confident regions and stricter in ambiguous ones, thereby reducing inference time with minimal performance loss. For detailed metrics and comparisons, please refer to Sec. 4.3 and Sec. 4.4.

4 Experiments

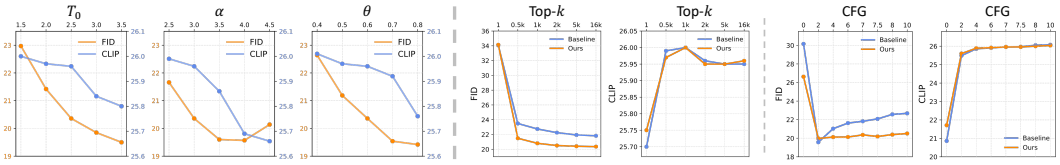
4.1 Implementation details

Four representative models are selected for comparison: vanilla AR model LlamaGen [1] and Lumina-mGPT [2] based on next-token prediction, mask-based model Meissonic [44], and scale-wise model STAR [13]. All models are evaluated under their original inference settings (e.g., CFG=4 and top-K=2000 for Lumina-mGPT, CFG=7.5 for LlamaGen). We use LlamaGen's official Stage-1 model to evaluate the sampling strategy, while Stage-2 is used only for acceleration analysis due to its poor performance (FID 53.42, CLIP-Score 21.47), which makes quality differences hard to observe.

FID and CLIP-Score are tested on the MS-COCO 2017 [64] validation set to evaluate the image quality and prompt-following capability. Moreover, DPG-bench [65] and HPS [66] are adopted to assess the semantic fidelity and perceptual quality of the generated images. All experiments are conducted on A100 GPUs.

4.2 Sampling quality

As shown in Table 1 and Fig. 5–6, our dynamic temperature sampling strategy effectively adapts to regions with varying information density in the image, leading to more stable structures and clearer details in the generated outputs. Depending on the inherent sampling mechanism of each model, our method yields varying degrees of improvement across different approaches. In particular, it achieves an approximate 4-point gain on DPG for both Meissonic and LlamaGen, along with a notable enhancement in visual quality. In addition, integrating our approach with the masking- and scale-wise strategies described in Sec. 3.3 can further enhance generation performance. See Table 1 for results with "+Prob" (applying dynamic temperature to logits only), and "+Masking / +Scale-wise" (applying temperature based on mask or scale).



(a) Ablation of Parameters in Eq. (2)

(b) Combination of Proposed Method with top-k and CFG

Figure 8: (a) Impact of parameters in Sec. 3.2; (b) Combination of our sampling strategy with existing methods (Top-K, CFG). Our method reduces the sensitivity of FID and CLIP-Score to these parameters, enhancing image quality and text alignment. Experiments are conducted on LlamaGen.

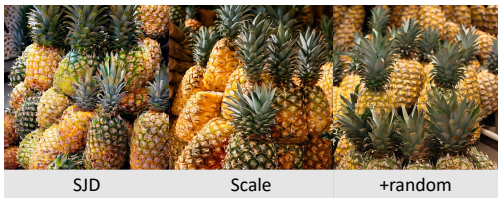


Figure 9: Visual comparison for Table 3. "Scale" slightly degrades quality while accelerating, which can be mitigated by +random".

Table 3: Quantitative comparison on acceleration. "Avg. Lat." is short for "Avg. Latency." Experiments are conducted on Lumina-mGPT.

Config.	Avg. Lat.↓	Avg. NFE↓	FID↓	CLIP Score↑
SJD [26]	84.97	1854.5	30.76	26.09
scale	69.38	1523.7	33.86	26.05
+random	72.35	1594.5	30.89	26.12

4.3 Inference acceleration

By integrating with existing vision-based speculative decoding schemes and leveraging entropy to automatically control the acceptance condition, our method saves about 15% inference cost with almost no loss in image generation quality compared to the approach in [26], as shown in Table 2 and Fig. 7. The entropy-based approach significantly reduces the number of inference steps and latency, while still maintaining comparable image quality to the original speculative decoding method.

4.4 Ablation study and discussion

Parameters in Eq. 2. The impact of parameters in entropy-aware dynamic temperature is provided in Fig. 8(a). It is observed that smaller ϵ or θ values lead to higher FID and lower CLIP-Score, primarily due to decreased randomness and overly deterministic sampling. Meanwhile, FID shows a trend of initially decreasing and then increasing as α increases. This is because α governs the proportion of different temperatures. When α is too small, most tokens are assigned very low temperatures, causing the FID to increase. Conversely, when α is too large, the image content becomes overly chaotic, resulting in increased FID. CLIP-Score consistently decreases as α increases.

Acceptance rate in Sec. 3.4. We propose to dynamically control the acceptance rate in existing speculative decoding methods based on the entropy of predicted distributions. By adjusting both the scale and randomness of the threshold r, we reduce latency while maintaining quality. As shown in Table 3 and Fig. 9, controlling only scale of r ("scale") reduces inference cost but degrades performance, especially image quality. In contrast, jointly tuning both scale and randomness ("+random") achieves a better trade-off, enabling high-quality generation with minimal inference overhead.

Compatibility with different AR models. Our sampling method brings notable performance gains for some models—for instance, DPG in LlamaGen and Meissonic outperforms baseline by over 3 points. In contrast, well-trained models like Lumina-mGPT benefit only marginally. This discrepancy stems from factors such as generation paradigm (*e.g.*, inherent sampling limitations of mask-based methods discussed in Fig. 4), training datasets and iterations (*e.g.*, whether has been thoroughly trained on large-scale data). Nevertheless, these models can still exploit entropy for further acceleration.

Combination with top-K and CFG. We further analyze the performance of our method when combined with top-K sampling and CFG, as shown in Fig. 8(b); results with top-p and temperature are in the supplementary. By incorporating proposed method, FID metric becomes less sensitive to sampling parameters, enabling better fidelity while maintaining image-text alignment.

Factors affecting entropy. Unlike text generation with fixed tokenization rules, autoregressive image generation relies on pretrained tokenizers, and the underlying model differences—including

Figure 10: For some cases, the semantic-corresponding foreground contents may have smaller entropy.

Table 4: Mean and variance ("Var.") of entropy from [1] on COCOval17.

Reso.	CFG	Mean	Var.
256	7.5	4.76	2.75
256	4.0	5.90	2.42
256	2.0	6.87	1.97
512	7.5	4.42	3.09

parameter scale, data quality, and training corpus—lead to varying entropy distributions and optimal sampling parameters. Empirically, higher CFG and larger resolutions lead to lower average entropy (see Table 4). Further analyses are provided in supplementary material.

Discussion of failure cases. Although our entropy-based dynamic sampling strategy brings notable performance improvements, we also observe several failure cases where the relationship between semantic information and the entropy map becomes less consistent (see Fig. 10). In some cases, regions such as human faces exhibit unexpectedly high entropy, while complex backgrounds receive lower entropy values. Consequently, adjusting temperature based on such entropy patterns may lead to structural distortions and overly smooth details. This ambiguity may potentially limit further performance gains, especially for models that have been carefully optimized.

5 Conclusion

In this work, we first point out the need for different sampling strategies in autoregressive image and text generation, given their distinct information distributions. Starting from this perspective, we find entropy effectively represents image information density, offering new possibilities for improving and accelerating image generation. As our method involve parameter adjustments without training, this approach could be further integrated into training or fine-tuning frameworks, potentially accelerating training, boosting inference speed, improving stability, and reducing hyperparameter dependence.

6 Acknowledgments

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

References

[1] 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] Dongyang Liu, Shitian Zhao, Le Zhuo, Weifeng Lin, Yu Qiao, Hongsheng Li, and Peng Gao. Lumina-mgpt: Illuminate flexible photorealistic text-to-image generation with multimodal generative pretraining. *arXiv preprint arXiv:2408.02657*, 2024.

[3] 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.

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

[5] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[6] Qwen Team et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(8), 2024.

[7] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in Neural Information Processing Systems*, 37:84839–84865, 2024.

[8] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12966–12977, 2025.

[9] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

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

[11] 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.

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

[13] Xiaoxiao Ma, Mohan Zhou, Tao Liang, Yalong Bai, Tiejun Zhao, Huaian Chen, and Yi Jin. Star: Scale-wise text-to-image generation via auto-regressive representations. *arXiv preprint arXiv:2406.10797*, 2024.

[14] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022.

[15] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[16] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[17] 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. *arXiv preprint arXiv:2403.03206*, 2024.

[18] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

[19] Chenxia Tang, Jianchun Liu, Hongli Xu, and Liusheng Huang. Top-$n\sigma$: Not all logits are you need. *arXiv preprint arXiv:2411.07641*, 2024.

[20] Shimao Zhang, Yu Bao, and Shujian Huang. Edt: Improving large language models' generation by entropy-based dynamic temperature sampling. *arXiv preprint arXiv:2403.14541*, 2024.

[21] Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. A contrastive framework for neural text generation. *Advances in Neural Information Processing Systems*, 35:21548–21561, 2022.

[22] Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*, 2023.

[23] Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13418–13427, 2024.

[24] Ziqin Luo, Haixia Han, Haokun Zhao, Guochao Jiang, Chengyu Du, Tingyun Li, Jiaqing Liang, Deqing Yang, and Yanghua Xiao. Sed: Self-evaluation decoding enhances large language models for better generation. *arXiv preprint arXiv:2405.16552*, 2024.

[25] Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*, 2025.

[26] Yao Teng, Han Shi, Xian Liu, Xuefei Ning, Guohao Dai, Yu Wang, Zhenguo Li, and Xihui Liu. Accelerating auto-regressive text-to-image generation with training-free speculative jacobi decoding. In *The Thirteenth International Conference on Learning Representations*.

[27] Hongyang Wei, Shuaizheng Liu, Chun Yuan, and Lei Zhang. Perceive, understand and restore: Real-world image super-resolution with autoregressive multimodal generative models. *arXiv preprint arXiv:2503.11073*, 2025.

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

[29] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021.

[30] Yuying Ge, Yixiao Ge, Ziyun Zeng, Xintao Wang, and Ying Shan. Planting a seed of vision in large language model. *arXiv preprint arXiv:2307.08041*, 2023.

[31] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. Pmlr, 2021.

[32] 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*, 2(3):5, 2022.

[33] Wanggui He, Siming Fu, Mushui Liu, Xierui Wang, Wenyi Xiao, Fangxun Shu, Yi Wang, Lei Zhang, Zhelun Yu, Haoyuan Li, et al. Mars: Mixture of auto-regressive models for fine-grained text-to-image synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 17123–17131, 2025.

[34] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

[35] Ethan Chern, Jiadi Su, Yan Ma, and Pengfei Liu. Anole: An open, autoregressive, native large multimodal models for interleaved image-text generation. *arXiv preprint arXiv:2407.06135*, 2024.

[36] Yang Jiao, Haibo Qiu, Zequn Jie, Shaoxiang Chen, Jingjing Chen, Lin Ma, and Yu-Gang Jiang. Unitoken: Harmonizing multimodal understanding and generation through unified visual encoding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3600–3610, 2025.

[37] Chuofan Ma, Yi Jiang, Junfeng Wu, Jihan Yang, Xin Yu, Zehuan Yuan, Bingyue Peng, and Xiaojuan Qi. Unitok: A unified tokenizer for visual generation and understanding. *arXiv preprint arXiv:2502.20321*, 2025.

[38] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. *arXiv preprint arXiv:2406.07550*, 2024.

[39] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11523–11532, 2022.

[40] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

[41] Yue Zhao, Yuanjun Xiong, and Philipp Krähenbühl. Image and video tokenization with binary spherical quantization. *arXiv preprint arXiv:2406.07548*, 2024.

[42] Guiwei Zhang, Tianyu Zhang, Mohan Zhou, Yalong Bai, and Biye Li. V2flow: Unifying visual tokenization and large language model vocabularies for autoregressive image generation. *arXiv preprint arXiv:2503.07493*, 2025.

[43] Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K Du, Zehuan Yuan, and Xinglong Wu. Tokenflow: Unified image tokenizer for multimodal understanding and generation. *arXiv preprint arXiv:2412.03069*, 2024.

[44] Jinbin Bai, Tian Ye, Wei Chow, Enxin Song, Qing-Guo Chen, Xiangtai Li, Zhen Dong, Lei Zhu, and Shuicheng Yan. Meissonic: Revitalizing masked generative transformers for efficient high-resolution text-to-image synthesis. In *The Thirteenth International Conference on Learning Representations*, 2024.

[45] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

[46] Hu Yu, Biao Gong, Hangjie Yuan, DanDan Zheng, Weilong Chai, Jingdong Chen, Kecheng Zheng, and Feng Zhao. Videomar: Autoregressive video generatio with continuous tokens. *arXiv preprint arXiv:2506.14168*, 2025.

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

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

[49] Hu Yu, Hao Luo, Hangjie Yuan, Yu Rong, and Feng Zhao. Frequency autoregressive image generation with continuous tokens. *arXiv preprint arXiv:2503.05305*, 2025.

[50] Yefei He, Yuanyu He, Shaoxuan He, Feng Chen, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. Neighboring autoregressive modeling for efficient visual generation. *arXiv preprint arXiv:2503.10696*, 2025.

[51] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. *arXiv preprint arXiv:2411.00776*, 2024.

[52] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[53] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.

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

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

[56] Clara Meister, Tim Vieira, and Ryan Cotterell. If beam search is the answer, what was the question? *arXiv preprint arXiv:2010.02650*, 2020.

[57] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

[58] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

[59] Doohyuk Jang, Sihwan Park, June Yong Yang, Yeonsung Jung, Jihun Yun, Souvik Kundu, Sung-Yub Kim, and Eunho Yang. Lantern: Accelerating visual autoregressive models with relaxed speculative decoding. In *The Thirteenth International Conference on Learning Representations*.

[60] Yefei He, Feng Chen, Yuanyu He, Shaoxuan He, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. Zipar: Accelerating autoregressive image generation through spatial locality. *arXiv preprint arXiv:2412.04062*, 2024.

[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.

[62] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.

[63] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2023.

[64] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014.

[65] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

[66] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

[67] Yufei Wang, Lanqing Guo, Zhihao Li, Jiaxing Huang, Pichao Wang, Bihan Wen, and Jian Wang. Training-free text-guided image editing with visual autoregressive model. *arXiv preprint arXiv:2503.23897*, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: NA

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: NA

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: NA

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: NA

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     1. If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     2. If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     3. If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     4. We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: NA

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: NA

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: NA

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: NA

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: NA

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper does not release new assets.

- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Introduction

We first provide additional details of our method, including parameter settings and further descriptions, in Sec. B. Then, we present extended experimental results in Sec. C. In Sec. D, we conduct deeper analyses on entropy in relation to model behavior and image content, along with more visualizations of entropy maps. Sec. E discusses potential limitations and future directions. Lastly, we include more visual comparisons of the proposed method in Sec. F.

B Additional details of our method

B.1 Hyperparameter settings in Sec. 3.2

In Sec. 3.2, we propose to dynamically control the sampling temperature based on entropy. However, due to significant differences between base models, it is difficult to apply the same parameters across all settings. Therefore, we list the detailed parameters for each model in Table 5. For undertrained models such as LlamaGen stage1, higher randomness is required at low-entropy stages to avoid generating large areas of repetitive tokens. In contrast, well-trained models benefit from a smoother temperature schedule.

Table 5: Hyperparameter settings of different models.

	T_0	α	θ
LlamaGen	2.5	3.0	0.6
Lumina-mGPT	2.0	2.5	0.6
Meissonic	2.5	3.0	0.7
STAR	2.5	3.0	0.5

B.2 Detailed description of speculative decoding in images

We accelerate inference based on existing speculative decoding schemes [26] in Sec. 3.4, thereby further reducing inference cost without sacrificing output quality. Due to space constraints, we did not elaborate on the baseline speculative decoding methods in the main text. Here, we provide more details.

This method aims to accelerate auto-regressive text-to-image generation by allowing multiple tokens to be generated in parallel without training. Inspired by speculative decoding, SJD introduces a probabilistic acceptance criterion that compares the confidence of draft tokens from two consecutive iterations. In each iteration j, given a draft token $x_i^{(j)}$, SJD computes its acceptance probability based on the ratio between two conditional probabilities:

$$r < \min\left(1, \frac{p_\theta(x_i^{(j)} \mid x_{1:i-1}^{(j)})}{p_\theta(x_i^{(j)} \mid x_{1:i-1}^{(j-1)})}\right), \tag{9}$$

where $r \sim \mathcal{U}[0,1]$. Accepted tokens are fixed, while the others are resampled from a calibrated distribution:

$$x_i^{(j+1)} \sim \frac{\max(0, p_\theta(x \mid x_{1:i-1}^{(j)}) - p_\theta(x \mid x_{1:i-1}^{(j-1)}))}{\sum_x \max(0, \cdot)}. \tag{10}$$

This allows high-randomness sampling, crucial for image diversity, while significantly reducing decoding steps. SJD operates in a windowed, iterative manner and supports optional spatially-informed token initialization to further improve efficiency.

C Additional experimental results

C.1 Generation performance on an additional dataset

Since the COCO2017 dataset used in the main experiments contains only 5,000 images, it may lead to slight estimation bias in the FID computation, as FID becomes more reliable with larger sample sizes. To assess the potential misjudgment of model performance caused by limited image numbers, we further evaluated the metrics on a larger dataset, COCO2014, as shown in Table 6.

Table 6: Evaluation on the larger COCO2014 dataset (compared to COCO2017 in the main text). The results demonstrate that the improvements brought by our method remain consistent and significant across datasets.

Model	Method	FID↓	CLIP-Score↑
LlamaGen [1]	Baseline	13.37	**0.2561**
	Ours	**11.59**	0.2560
Meissonic [44]	Baseline	44.54	0.2567
	Ours+Mask	**38.95**	**0.2590**
STAR [13]	Baseline	24.86	0.2581
	Ours+Scale	**22.09**	**0.2598**
Lumina-mGPT [2]	Baseline	18.23	0.2641
	Ours	**16.16**	**0.2659**

C.2 Effect of random seeds on model performance

Due to the autoregressive nature of our model, each token is sampled from a probability distribution, making the generated images sensitive to random seeds. Specifically, we observed that metrics such as FID, CLIP-Score, DPG, and HPS may vary with different random seeds. To further analyze this effect, we randomly selected 10 seeds from the range $[0, 1e6]$, ran the generation model 10 times under these conditions, and computed the mean and standard deviation of the results. As shown in the table below, random seeds have little impact on the performance gain introduced by our method, further confirming the robustness and effectiveness of our approach. See Table 7.

Table 7: Mean and standard deviation over 10 random seeds. Our method consistently outperforms the baseline with statistically significant improvements.

	LlamaGen		Meissonic		STAR	
	Baseline	Ours	Baseline	Ours	Baseline	Ours
FID↓	21.79 ± 0.16	20.24 ± 0.08	53.31 ± 0.33	48.18 ± 0.29	35.48 ± 0.29	33.12 ± 0.49
CLIP↑	25.95 ± 0.02	25.95 ± 0.03	25.30 ± 0.02	25.62 ± 0.03	25.47 ± 0.02	25.63 ± 0.03
DPG↑	43.74 ± 0.34	48.87 ± 0.32	64.07 ± 0.13	66.91 ± 0.17	70.28 ± 0.12	70.40 ± 0.11
HPS↑	21.23 ± 0.03	21.40 ± 0.04	29.33 ± 0.03	30.06 ± 0.02	28.70 ± 0.08	29.07 ± 0.11

C.3 Entropy & top-p and temperature

In the main text, we analyze the relationship between our entropy-based sampling strategy and existing sampling parameters such as CFG and top-K. By combining our method with these parameters, we observe improved robustness, reducing sensitivity to hyperparameter choices and yielding better FID and CLIP-Score. Here, we further examine other sampling parameters—top-p and temperature—which are rarely used in autoregressive models due to their tendency to distort the output distribution and severely degrade either FID or CLIP-Score. Comparative results between our method and the baseline are shown in Fig. 11.

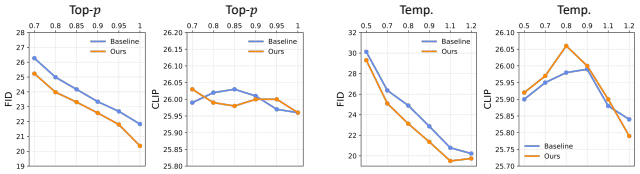


Figure 11: Combination of our sampling strategy with existing methods (Top-p, temperature). "Temp." is short for temperature.

C.4 Additional comparison with top-K and CFG

In Sec. 4.4 of the main paper, we discussed the differences between our method and existing sampling strategies (Top-K and CFG) using LlamaGen. Here, we provide additional comparisons on other models to analyze the relationship between entropy-aware temperature and these conventional sampling approaches, results are presented in Fig. 12 and Fig. 13. Consistent with our observations in

Sec. 4.4, the proposed strategy mitigates performance fluctuations caused by hyperparameter choices (e.g., CFG and top-K), leading to a better balance between fidelity and text-image alignment.

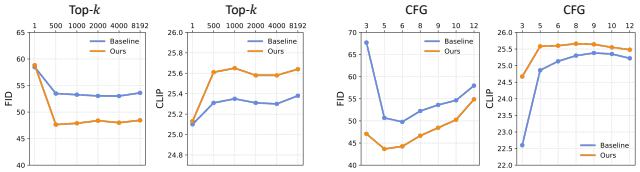


Figure 12: Combination of our sampling strategy with existing methods (Top-k, CFG) on Meissonic.

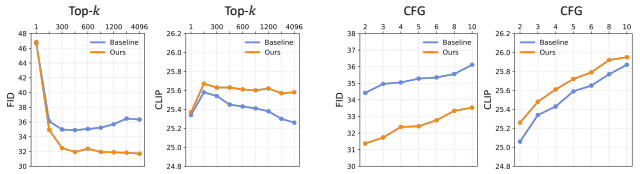


Figure 13: Combination of our sampling strategy with existing methods (Top-k, CFG) on STAR.

C.5 Additional evaluation of performance regarding temperature

In the main text, we analyzed how adjusting the sampling temperature of tokens in different entropy ranges affects the generation quality for LlamaGen. Here, we further extend the study to more models. See Fig. 14.

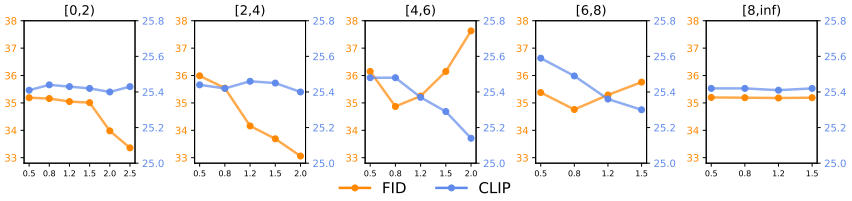


Figure 14: Varying temperature by entropy range affects FID and CLIP score: lower-entropy tokens benefit from higher temperatures, and vice versa. Experiments are conducted on STAR using the COCO2017 validation split by varying temperature across entropy ranges.

D Additional discussion about entropy

D.1 Entropy & generative models

D.1.1 Visualization of entropy & images

Due to space limitations in the main text, we did not provide extended entropy visualizations and analysis. Here, we include additional entropy maps for LlamaGen and Lumina-mGPT. See Fig. 15 and Fig. 16.

D.1.2 Mask-prediction models

We provide additional entropy-based analysis of the mask model. Since the generation involves multiple timesteps, where a subset of tokens is accepted at each step based on previously generated content, we compute the entropy of accepted tokens at each timestep and aggregate them into a final entropy map. As shown in Fig. 17, applying the proposed entropy-based temperature leads to a more spatially balanced entropy distribution and enables richer image content while maintaining generation stability.

In addition, we further analyze the average entropy of tokens accepted at each timestep, as shown in Fig. 18. As discussed in the main text, due to the confidence-based token selection strategy, tokens accepted in earlier steps tend to have lower entropy, since they are more likely to receive high

Figure 15: Entropy visualization of LlamaGen.

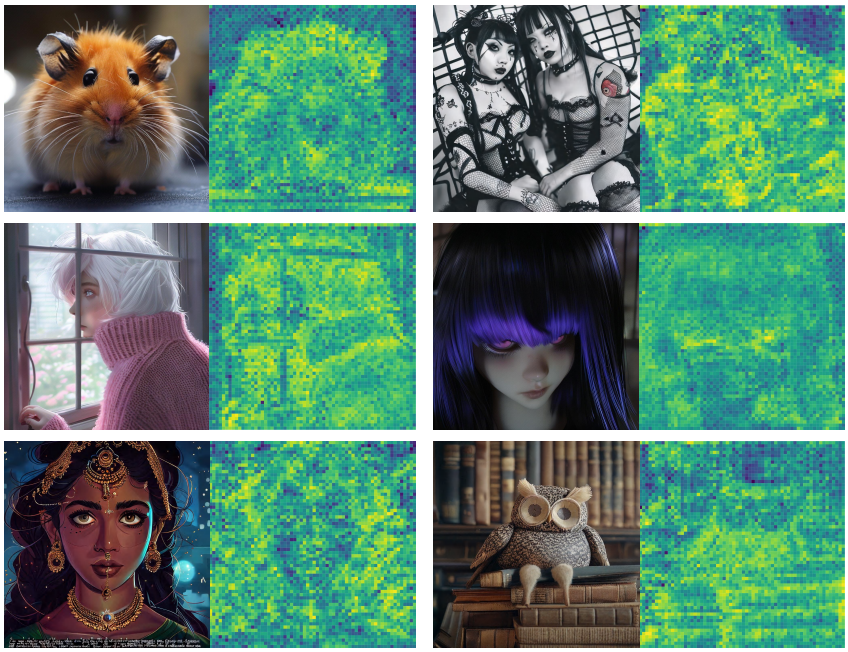


Figure 16: Entropy visualization of Lumina-mGPT.

confidence scores. In contrast, tokens accepted in later steps (>60) exhibit significantly higher entropy. Moreover, more tokens are accepted in these later stages, which increases the risk of violating the autoregressive assumption that spatially adjacent tokens should be sampled as independently as possible. This may lead to degraded image quality. Therefore, adopting a more conservative sampling strategy for these high-entropy tokens could help improve the overall generation quality.

D.1.3 Scale-wise models

For the scale-wise model, the generation process constructs a complete image by predicting logits maps at multiple scales. Each scale is conditioned on the residuals from the preceding scales, meaning that the sum of the feature maps generated at all scales is passed through the detokenizer to form the final output. In this generation paradigm, different scales exhibit distinct roles. Specifically, as described in [67], the earlier scales are responsible for generating the main structure of the image, while the later scales refine the result with fine details such as texture. We visualize the entropy maps of each scale during generation, as shown in Fig. 19. From scale 8 to scale 12, the model

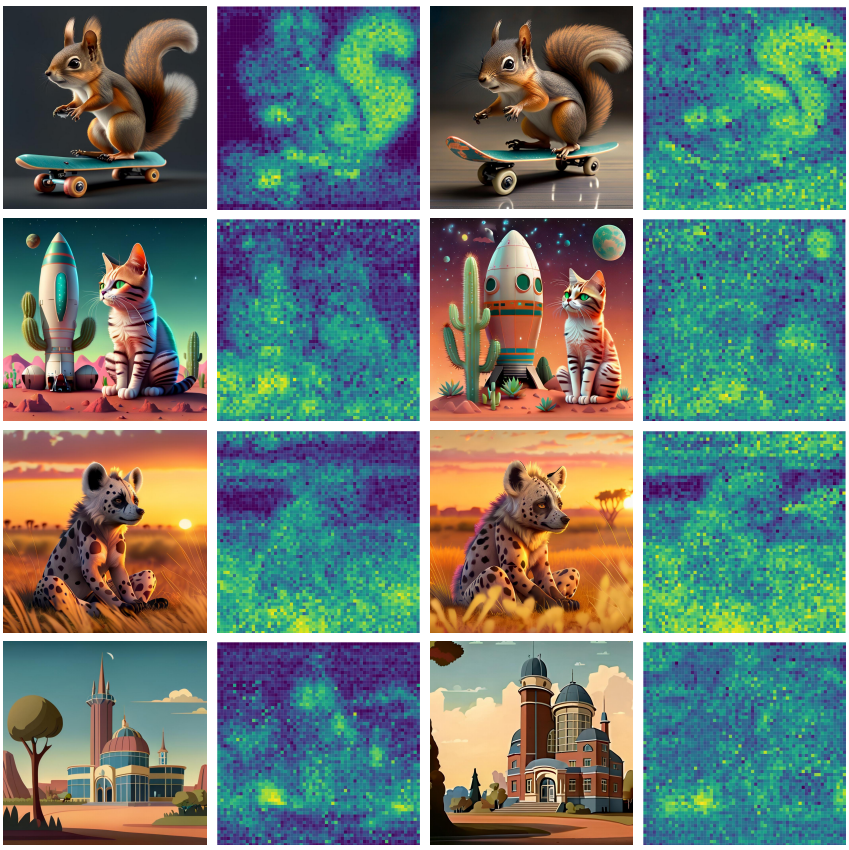


Figure 17: Entropy visualization of Meissonic. Our entropy-based temperature leads to a more spatially balanced entropy distribution and enables richer image content.

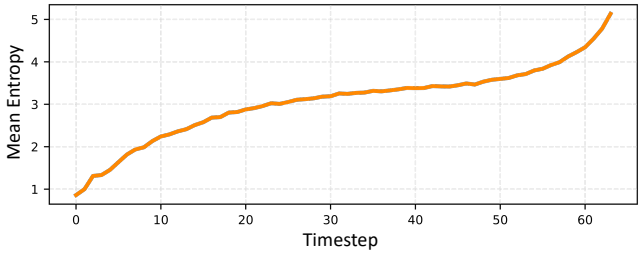


Figure 18: Mean entropy of each step from mask-prediction model. Values are averaged from ∼100 generated images.

tends to focus more on the foreground, with significantly higher entropy observed in the regions corresponding to the primary subject. In contrast, at scale 13 and 14, there is no clear bias between foreground and background, indicating a more uniform attention across the image.

In addition, we compute the average entropy for each scale, as shown in Fig. 20. The later scales exhibit relatively higher entropy, while the earlier scales tend to have lower average entropy (however a decreasing trend is observed in the final two scales). This further indicates that different scales carry varying amounts of information.

D.2 Entropy & generated contents

In practice, the logits are not simply positively correlated with the complexity of image content. We observe that regions with clear, well-defined content do not always exhibit high entropy; instead, their

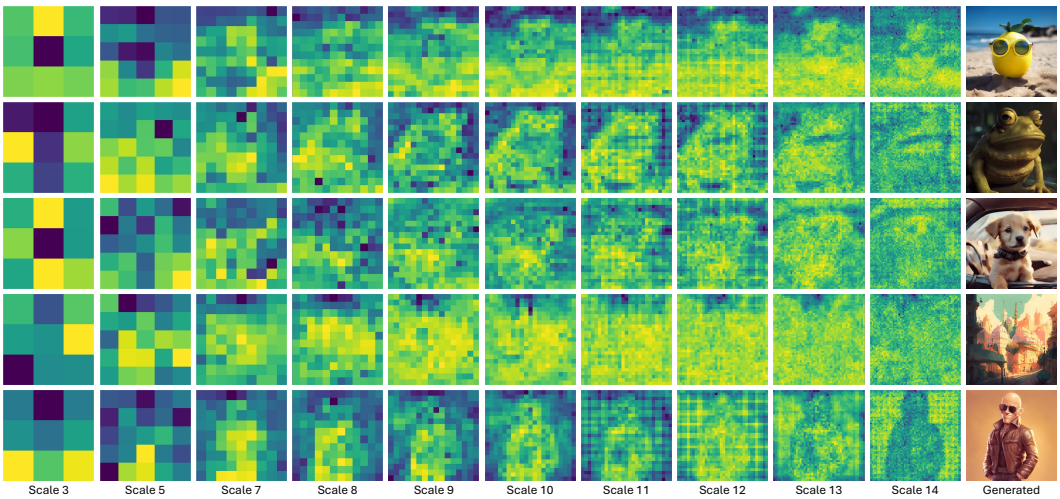


Figure 19: Entropy visualization of STAR. .

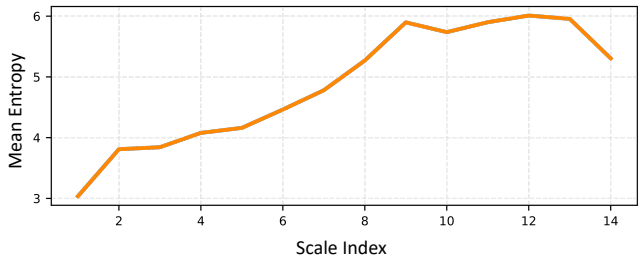


Figure 20: Mean entropy of each scale from scale-wise model. Values are averaged from ∼100 generated images.

entropy typically falls within a moderate range (e.g., between 2 and 8). The more deterministic the content, the lower the entropy tends to be. In contrast, regions with entropy lower than 2 or higher than 8 often correspond to simple backgrounds or overly complex, unfaithful details. Especially for regions with entropy above 8, the generated details are frequently meaningless. This also explains why adjusting the logits in these low- and high-entropy areas, as discussed in our motivation experiment, does not significantly harm text-image alignment.

D.3 Is entropy the best indicator for information?

From a theoretical perspective, the entropy of logits reflects the model's confidence in predicting the current token. When the model is sufficiently trained—or when its capacity is strong—it may produce low entropy even in semantically important regions. In practice, we observe cases where foreground objects (e.g., faces) yield lower entropy than complex backgrounds. This suggests that the model's confidence is not solely determined by information density, but also by the number of plausible token candidates in a region. For instance, highly structured areas like faces tend to have a unique correct token and thus low uncertainty, despite containing rich semantic information. In contrast, cluttered textures such as grass or foliage may allow for more varied token predictions, resulting in higher entropy.

Based on the above analysis, entropy may need to be combined with additional indicators to more accurately characterize the information distribution within an image. Specifically, more precise token-wise handling can be achieved by incorporating the similarity among top-ranked tokens in the logits distribution. For instance, if the entropy is low but the top tokens are not similar, the prediction can be deemed accurate; however, if the top tokens are highly similar under low entropy, the randomness at that position may need to be further increased. Conversely, under high-entropy conditions, a set of similar top tokens may indicate the existence of genuinely diverse possibilities. Moreover, analyzing the similarity of logits between adjacent tokens could help identify tokens that

require more precise predictions—for example, if a token's probability distribution significantly differs from that of the previous token, it may warrant stricter sampling, regardless of its entropy level. We leave these directions for future exploration.

Moreover, since dynamic temperature only adjusts the randomness of the probability distribution (i.e., the variance of the logits) but not the location of its peak, further combining it with CFG may help achieve better performance.

E Future works & limitations

E.1 Broader impacts

This work is the first to explore the decoding problem in autoregressive visual generation, highlighting the inherent differences between image and text generation. While our approach may not be fully complete and still leaves room for improvement, we hope it can inspire future research to further investigate this issue and develop decoding strategies tailored specifically for visual generation, ultimately advancing unified multimodal generation.

E.2 Future works

Currently, we propose a training-free sampling strategy for image generation by adaptively controlling sampling randomness based on the distribution of predicted logits. However, this approach is sensitive to hyperparameters, and due to significant differences across backbone architectures, optimal settings vary across models. Moreover, as a simple inference-time method built upon pretrained models, its performance gains may be limited for certain models.

In the future, this strategy could be integrated into the training framework for further performance improvement or acceleration. For example, it may be combined with early-exit mechanisms to allocate computation dynamically across tokens, or used to guide training by leveraging entropy to focus more on informative regions, thus accelerating convergence.

E.3 Limitations

The proposed method mainly mitigates issues caused by inconsistent token sampling strategies under varying information densities, but it does not enhance the intrinsic generation capability of autoregressive models. The performance gain is model-dependent. If the base model is trained with techniques that promote diverse token distributions, such as noise injection during training, or is well-trained on large-scale datasets, the improvement tends to be limited. Moreover, for weak base models, such as LlamaGen Stage 2, the method may offer little or no performance gain.

F Additional visual comparison

Due to space constraints in the main paper, we did not provide additional visualizations. Here, we include further results illustrating the entropy-aware sampling behavior for LlamaGen, Lumina-mGPT, Meissonic, and STAR, as well as acceleration visualizations for LlamaGen and Lumina-mGPT (see Fig. 21–Fig. 25).

Figure 21: Visualization of LlamaGen.

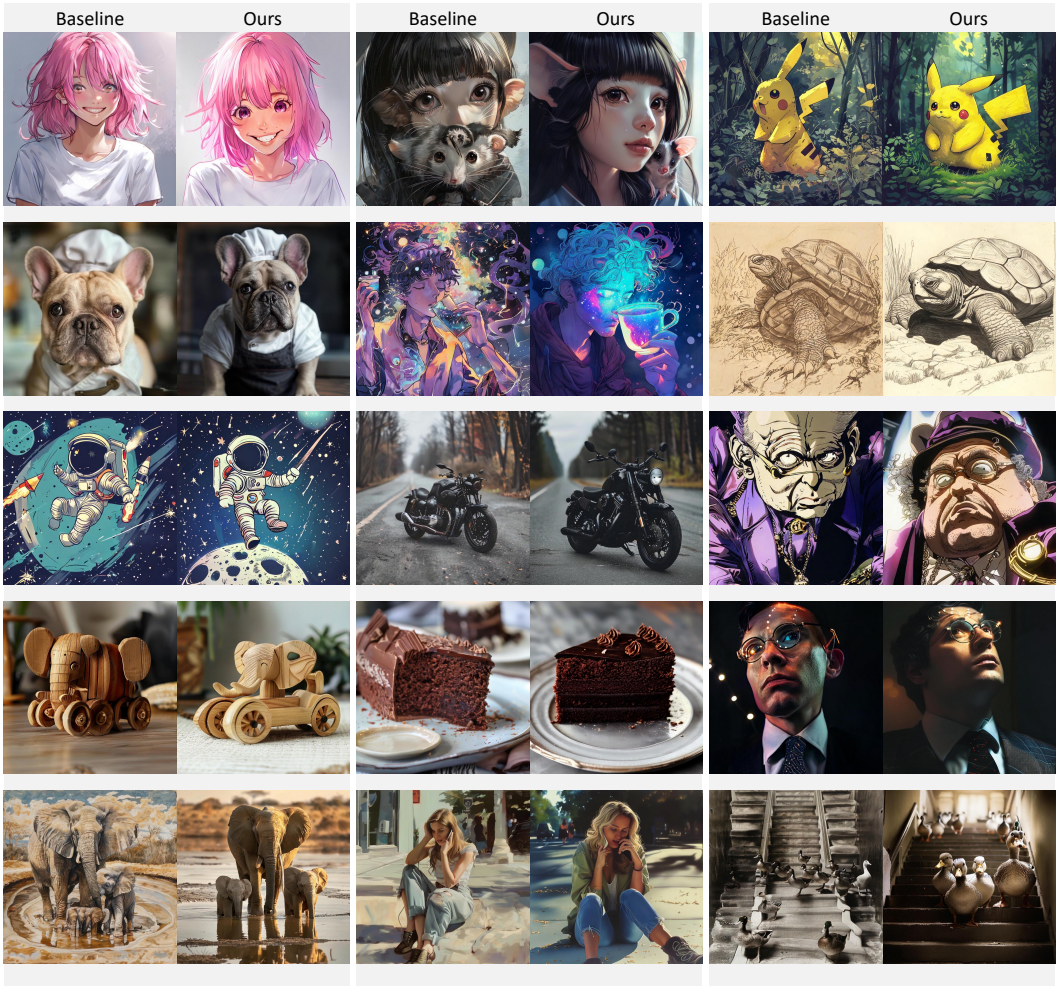


Figure 22: Visualization of Lumina-mGPT.

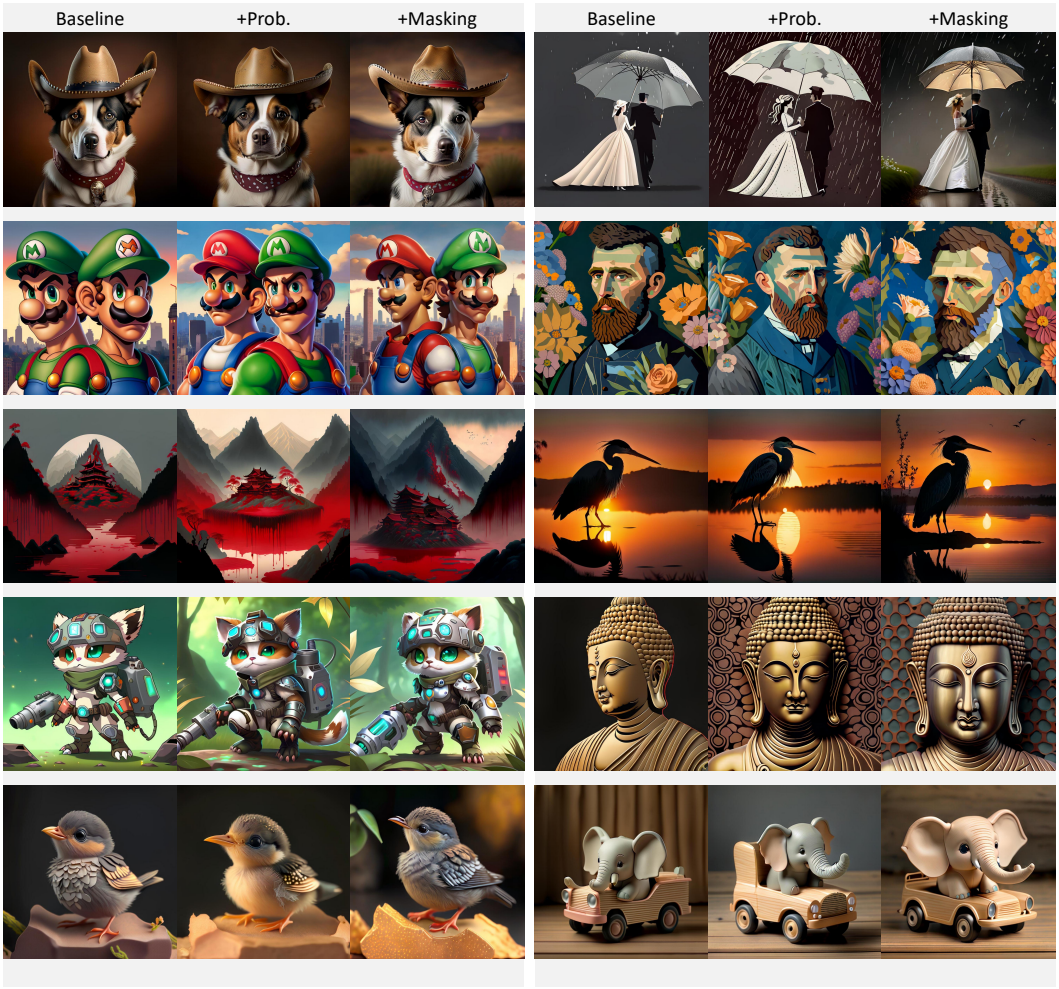


Figure 23: Visualization of Meissonic.

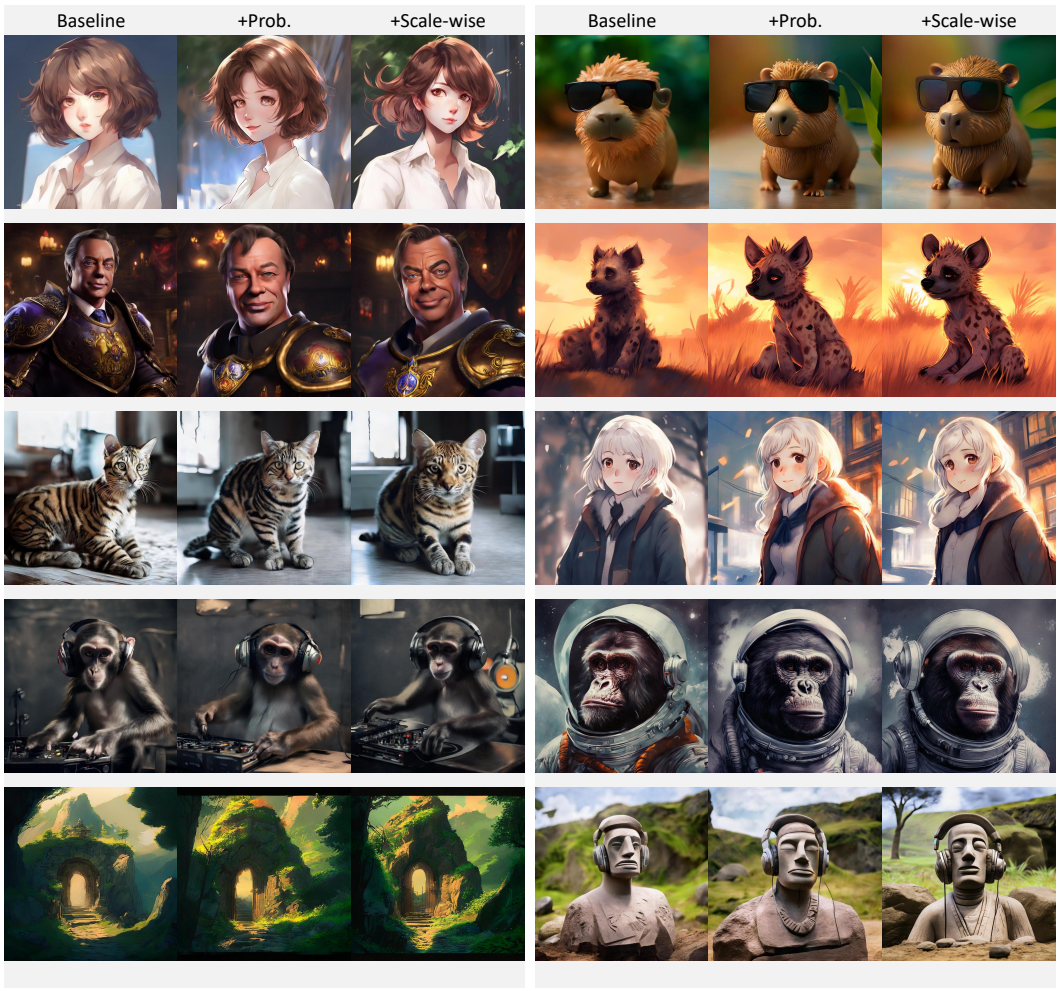


Figure 24: Visualization of STAR.

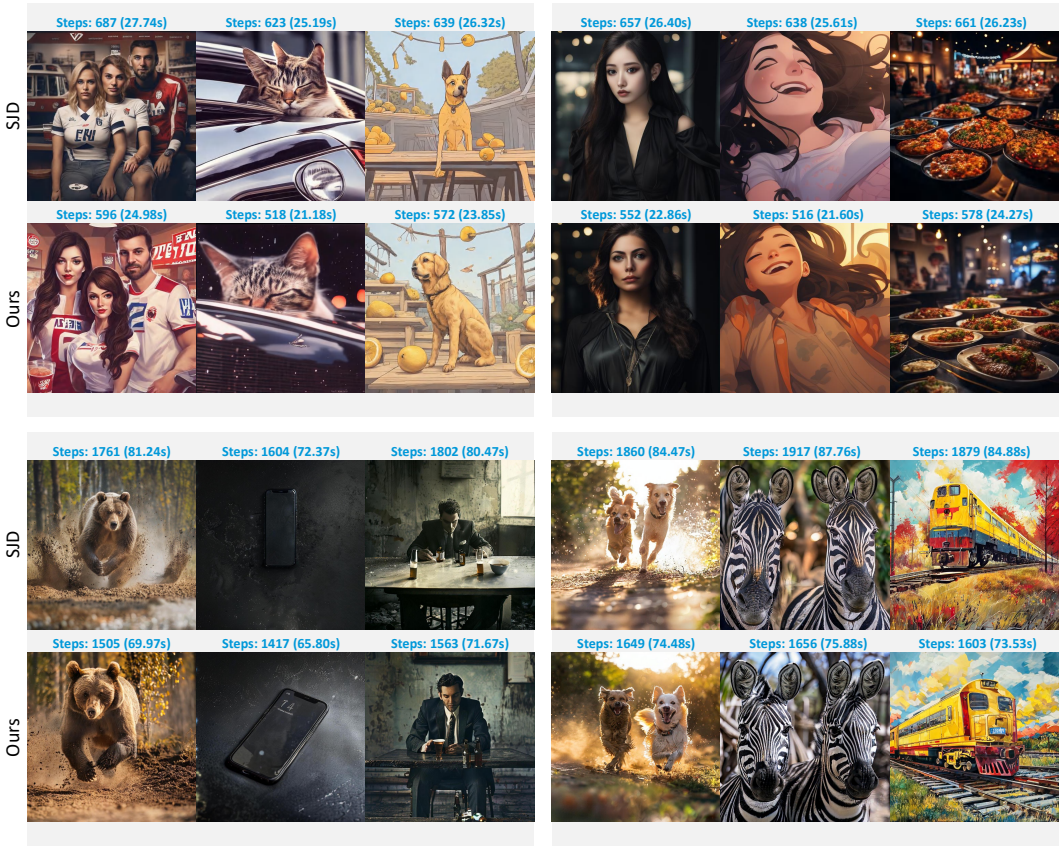


Figure 25: Visualization of AR acceleration.