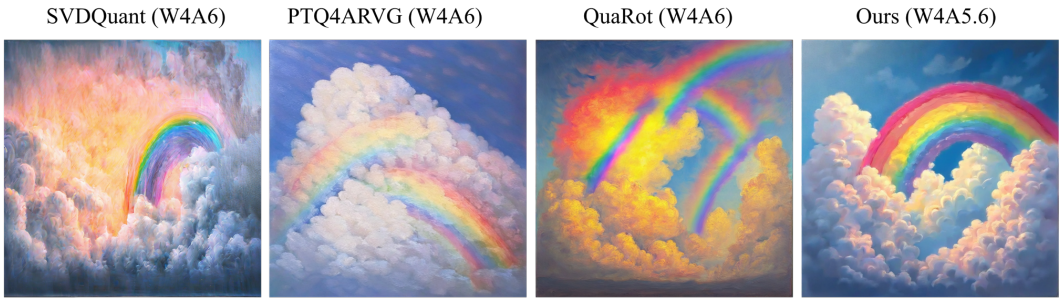
HGA-QUANT: HETEROGENEITY-AWARE QUANTIZATION FOR AUTOREGRESSIVE VISUAL GENERATION MODELS

Anonymous authors
Paper under double-blind review



Prompt: beautiful rainbow clouds, oil painting



Prompt: full body portrait of cute girl wearing gown, d &, fantasy, portrait, highly detailed, digital painting, illustration, art by artgerm and greg rutkowski and magali villeneuve

ABSTRACT

Recently, AutoRegressive Visual Generation (ARVG) models have emerged as a competitive paradigm for high-quality image synthesis. However, their substantial computational and memory demands make them difficult to deploy on resource-constrained edge devices. Post-Training Quantization (PTQ) provides a promising solution to reduce the computational and memory costs of ARVG models, yet existing methods often suffer from substantial image quality degradation under low-bit quantization. In this paper, we propose a training-free PTQ framework for ARVG, consisting of Tail-Aware Bounded-init Grid Refinement (T-BGR) and Coarse-to-Fine Activation Precision Scheduling (CAPS). T-BGR introduces a tail-aware reconstruction objective for weight quantizer calibration, assigning greater importance to distribution-tail coefficients and preserving fine-grained information without introducing additional parameters or high-precision branches. CAPS performs coarse-to-fine activation precision scheduling by jointly considering the importance and computational cost of different generation stages, enabling efficient mixed-precision allocation under a given bit budget. Extensive experiments on multiple ARVG backbones demonstrate that our framework consistently improves robustness under aggressive low-bit quantization. Notably, on Infinity, our framework achieves near full-precision image generation quality at an effective W4A5.6 activation precision, while existing methods at W4A6 still exhibit noticeable quality degradation or inconsistent performance.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Autoregressive visual generation (ARVG) models have recently emerged as a competitive paradigm for high-quality image synthesis. A representative line of work, Visual Autoregressive Modeling (VAR), reformulates image generation as next-scale prediction and constructs images from coarse to fine (Tian et al., 2024). Subsequent models have explored residual refinement, flexible scale schedules, and bitwise token prediction, as exemplified by HART (Tang et al., 2024), FlexVAR (Jiao et al., 2025), and Infinity (Han et al., 2025). These models provide a compelling alternative to diffusion-based generation, but their large Transformer backbones and repeated token prediction still impose substantial computational and memory demands. Such costs become particularly restrictive when ARVG models are deployed outside large-scale data-center settings.

Post-Training Quantization (PTQ) is an attractive way to reduce the cost of pretrained generative models without retraining them. By representing weights and activations with low-bit integers, PTQ can reduce the numerical precision required by the dominant linear operations while retaining the original model structure. However, directly applying generic PTQ methods to ARVG models often causes severe image-quality degradation at aggressive bit-widths. Recent studies have identified channel-wise outliers, token-wise activation dynamics, and calibration-distribution mismatch as important sources of this difficulty (Liu et al., 2026). Other methods mitigate quantization errors through equivalent transformations (Xiao et al., 2023), rotations (Ashkboos et al., 2024), or auxiliary low-rank compensation (Li et al., 2025a). Despite these advances, the quality of visual autoregressive generation can still collapse when both weights and activations are quantized to low precision, indicating that the non-uniform structure of ARVG quantization errors requires more specialized treatment (Moon & Ham, 2026).

We identify two complementary forms of heterogeneity that are especially relevant to low-bit ARVG quantization. First, weight coefficients are distributed unevenly across a large central mass and a much smaller tail region. Under an unweighted reconstruction loss, the central mass of a channel usually dominates the calibration objective, while tail coefficients contribute relatively little despite their larger magnitudes. Consequently, a calibration objective that minimizes average coefficient error may favor central-mass fidelity while leaving the tail reconstruction trade-off insufficiently controlled. Existing outlier-handling approaches often redistribute the quantization difficulty or preserve outliers through additional transformations or compensation paths. While these designs can improve numerical reconstruction, they do not directly expose how the tail-versus-central reconstruction trade-off should affect the calibration of the scale and zero-point.

Second, the effect of activation quantization can vary along the ARVG generation trajectory. In the coarse-to-fine process of VAR-style models, early stages establish the global structural context on which later stages condition, while later stages operate on denser token grids and mainly refine local content (Tian et al., 2024). Uniform activation precision therefore ignores the position-dependent sensitivity of the generation trajectory as well as its computational cost. Protecting all stages equally can allocate precision to late refinements, whereas assigning precision only according to token count may overlook the different consequences of perturbing early and late stages. These observations motivate a quantization framework that answers two different questions: which values in a weight distribution deserve greater reconstruction priority, and when along the generation trajectory should higher activation precision be used?

To address these challenges, we propose Heterogeneity-Aware Quantization for AutoRegressive Visual Generation models (HGA-Quant), a training-free PTQ framework composed of two complementary components. Tail-Aware Bounded-init Grid Refinement (T-BGR) extends bounded grid refinement with a tail-aware reconstruction objective. It identifies distribution-tail coefficients using channel statistics and assigns them larger weights during the search for the quantization interval, scale, and zero point. T-BGR keeps the standard quantized-weight representation while directing the calibration objective toward tail reconstruction. Coarse-to-Fine Activation Precision Scheduling (CAPS) exploits the generation-order prior of ARVG models. It estimates a stage-wise floating-point operations (FLOPs) proxy for the targeted linear projections and allocates the available high-precision budget to an early prefix of coarse stages, while using lower precision for the remaining fine stages. Thus, T-BGR targets non-uniform reconstruction priorities within the weight distribution, whereas CAPS targets non-uniform sensitivity along the generation trajectory.

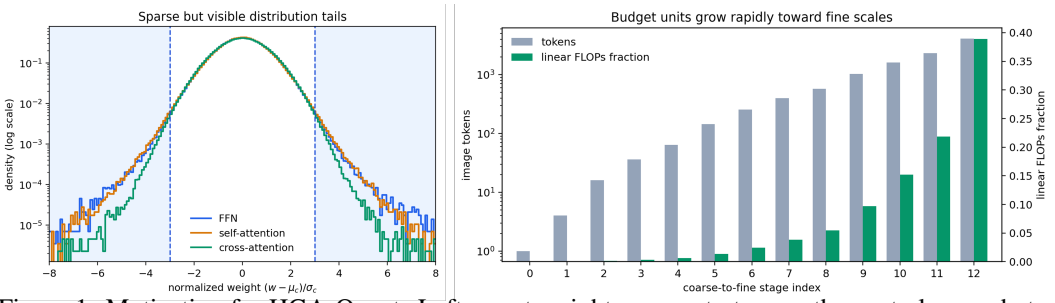


Figure 1: Motivation for HGA-Quant. Left: most weights concentrate near the central mass, but a non-negligible fraction lies in the tails, motivating tail-aware calibration. Right: token counts and linear-layer FLOPs grow sharply in later stages, motivating coarse-to-fine precision scheduling.

We evaluate HGA-Quant on four ARVG backbones, including Infinity-2B (Han et al., 2025), HART (Tang et al., 2024), VAR-d30 (Tian et al., 2024), and FlexVAR-d24 (Jiao et al., 2025), using W_bA_a to denote b-bit weights and a-bit activations, under W8A8, W4A8, W4A6, and W4A4 configurations. Across models and evaluation metrics, T-BGR remains competitive at higher precision and becomes increasingly robust relative to existing baselines as the bit-width decreases. On Infinity-2B, the stage-wise analysis in Figure 4 shows that the same A4 perturbation is substantially more damaging at early stages than at late stages. Accordingly, assigning W4A8 to the first 11 stages and W4A4 to the final two stages yields a FLOPs-weighted effective activation precision of 5.5711 bits, denoted W4A5.6 below, while maintaining performance comparable to uniform W4A6. The main tables report both uniform T-BGR (*Ours*) and the CAPS variant (*Ours (CAPS)*) where evaluated; the stage-wise and budgeted allocation studies further isolate the role of precision placement.

2 Related Works

2.1 AutoRegressive Visual Generation models

Recent ARVG models have emerged as a competitive paradigm for image synthesis. Representative approaches explore diverse autoregressive formulations, including scale-wise prediction (Tian et al., 2024), continuous-token modeling (Li et al., 2024b), randomized ordering (Yu et al., 2025), parallelized generation (Wang et al., 2024), and hybrid tokenization (Tang et al., 2024). VAR (Tian et al., 2024) is a foundational work that reformulates image generation as coarse-to-fine next-scale prediction, achieving strong image quality, inference efficiency, data efficiency, and scalability. HART (Tang et al., 2024) combines discrete autoregression with lightweight residual diffusion to improve reconstruction quality and enable efficient high-resolution generation. FlexVAR (Jiao et al., 2025) replaces residual prediction with ground-truth prediction at each scale, improving flexibility in resolution, aspect ratio, and inference steps. Infinity (Han et al., 2025) further scales ARVG through bitwise token prediction and an effectively enlarged token vocabulary. Its extensions have also explored unified image-video generation and image editing (Liu et al., 2025; Wang et al., 2025). Alongside architectural advances, recent studies have investigated efficient decoding (Chen et al., 2025; He et al., 2024), attention compression (Xie et al., 2024), and low-bit quantization (Xie et al., 2024; Wei et al., 2025) to reduce the computational and memory costs of ARVG models.

2.2 Post-Training Quantization

Quantization is a standard compression technique for large generative models. It reduces model size, memory traffic, and inference latency. PTQ enables efficient deployment without retraining. GPTQ (Frantar et al., 2022) uses Hessian-guided error compensation to minimize quantization error. AWQ (Lin et al., 2024) protects salient weights based on activation statistics. Methods based on scale redistribution and equivalent transformations (Shao et al., 2023; Xiao et al., 2023; Wei et al., 2023) mitigate activation outliers by shifting or redistributing quantization difficulty across weights and activations. Rotation-based methods (Ashkboos et al., 2024; Li et al., 2025b) further transform the representation space to disperse outliers across dimensions, facilitating low-bit quantization. SVDQuant (Li et al., 2025a) and Q-DiT4SR (Zhang et al., 2026) use a high-precision low-rank branch to compensate for quantization errors after outlier transformation. PTQ4ARVG (Liu et al., 2026) specifically studies PTQ for autoregressive visual generation, addressing channel-wise outliers, token-wise activation dynamics, and sample-wise distribution mismatch. Some recent works (Moon & Ham, 2026; Vezzali et al., 2026) have begun to explore PTQ and quantization specifically for visual autoregressive models, targeting different aspects of efficient deployment. De-

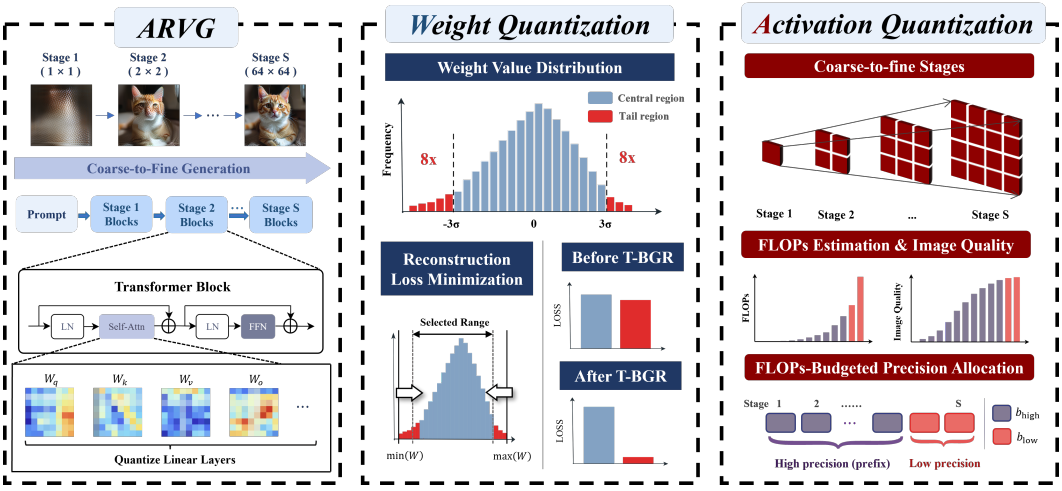


Figure 2: Overview of HGA-Quant. T-BGR performs tail-aware calibration for robust low-bit quantization, while CAPS assigns stage-wise precision in a coarse-to-fine manner for high-quality and efficient ARVG generation.

spite these advances, significant image quality degradation remains under aggressive quantization configurations such as 4-bit weights and 6-bit activations in visual autoregressive models.

3 METHOD

3.1 PRELIMINARIES

Autoregressive image generators such as Infinity synthesize an image through a sequence of coarse-to-fine stages. Let $\mathcal{S} = \{\mathbf{p}_i\}_{i=0}^{S-1}$ denote the scale schedule, where $\mathbf{p}_i = (p_{i,1}, \ldots, p_{i,D})$ specifies the token grid at stage i. The number of image tokens processed at this stage is $T_i = \prod_{d=1}^{D} p_{i,d}$. This generation process exposes two forms of non-uniformity that are not captured by conventional uniform post-training quantization (PTQ): different regions of a weight distribution can have different effects on the reconstructed output, and errors introduced at different generation stages need not propagate equally to the final image. We address these two sources of heterogeneity with T-BGR for tail-aware weight-quantizer calibration and CAPS for activation precision scheduling.

Post-training quantization converts a pretrained full-precision model to a low-bit model by determining quantization parameters from the weights and, when applicable, a small calibration set, without updating model parameters. For a scalar x, an unsigned b-bit asymmetric affine quantizer is written as

$$\mathcal{Q}_b(x; s, z) = s \left(\text{clip} \left(\left\lfloor \frac{x}{s} \right\rceil + z, 0, 2^b - 1 \right) - z \right), \tag{1}$$

Here, s is the scale and z is the zero point. The corresponding integer code is $q = \text{clip}(\lfloor x/s \rceil + z, 0, 2^b - 1)$, and the expression in Eq. equation 1 is its dequantized value.

3.2 TAIL-AWARE BOUNDED-INIT GRID REFINEMENT

3.2.1 MOTIVATION

Bounded-init Grid Refinement (BGR) searches candidate quantization intervals and refines the corresponding affine scale and zero point to reduce weight reconstruction error (Li et al., 2025b). Its conventional reconstruction objective weights all coefficients equally. Under this objective, the large central mass dominates the total error, while a small number of tail coefficients contributes relatively little to the loss despite their larger magnitudes. Therefore, preserving these coefficients might be important for retaining fine-grained responses under aggressive quantization.

We therefore propose *Tail-Aware Bounded-init Grid Refinement* (T-BGR), a tail-sensitive BGR extension. It modifies the calibration objective so that distribution-tail errors carry greater weight when selecting the quantization interval, scale, and zero point. This preserves the BGR deployment form while biasing its search toward a controlled tail-versus-central reconstruction trade-off.

3.2.2 TAIL-AWARE RECONSTRUCTION OBJECTIVE

Consider one output channel $\mathbf{w}_c \in \mathbb{R}^{C_{\text{in}}}$ of a weight matrix, where $w_{c,j}$ denotes its j-th weight coefficient. We compute its channel-wise mean and standard deviation as

$$\mu_c = \frac{1}{C_{\text{in}}} \sum_j w_{c,j}, \qquad \sigma_c = \sqrt{\frac{1}{C_{\text{in}}} \sum_j (w_{c,j} - \mu_c)^2}. \tag{2}$$

We identify a tail coefficient using a $\kappa\sigma$ criterion,

$$m_{c,j} = \mathbf{1}[|w_{c,j} - \mu_c| > \kappa\sigma_c], \tag{3}$$

and assign it the reconstruction weight

$$\rho_{c,j} = 1 + (\gamma - 1)m_{c,j}. \tag{4}$$

Here, κ controls the statistical tail boundary and γ controls the relative priority assigned to tail coefficients. A larger γ increases the influence of tail reconstruction on the selected quantization parameters, while a smaller κ identifies a larger portion of the channel as tail.

Let $\hat{w}_{c,j} = \mathcal{Q}_{b_w}(w_{c,j}; s_c, z_c)$ be the dequantized value of $w_{c,j}$, where b_w denotes the weight bit-width. T-BGR minimizes the normalized tail-aware reconstruction error

$$\mathcal{L}^{(c)}_{\text{T-BGR}}(s_c, z_c) = \frac{\sum_j \rho_{c,j} \left(\hat{w}_{c,j} - w_{c,j}\right)^2}{\sum_j \rho_{c,j}}. \tag{5}$$

When $\gamma = 1$, Eq. equation 5 reduces to the uniformly weighted reconstruction criterion used by the original BGR calibration. For $\gamma > 1$, the selected quantization parameters place greater emphasis on the reconstruction of tail coefficients, while the normalization keeps the loss scale comparable across channels with different tail ratios.

3.2.3 WEIGHTED GRID REFINEMENT

T-BGR retains the bounded grid-search structure of BGR. For each channel, we generate a set of candidate clipping intervals $[l_c^{(r)}, r_c^{(r)}]$ by progressively shrinking the original range $[\min_j w_{c,j}, \max_j w_{c,j}]$. For each candidate interval, we initialize an asymmetric affine quantizer and alternately update the scale and zero point. Given the current integer code $q_{c,j}$, the weighted scale update is

$$s_c \leftarrow \frac{\sum_j \rho_{c,j}(q_{c,j} - z_c)w_{c,j}}{\sum_j \rho_{c,j}(q_{c,j} - z_c)^2}, \tag{6}$$

followed by the weighted zero-point update

$$z_c \leftarrow \text{clip}\left(\left\lfloor \frac{\sum_j \rho_{c,j} \left(q_{c,j} - w_{c,j}/s_c\right)}{\sum_j \rho_{c,j}} \right\rceil, 0, 2^{b_w} - 1\right). \tag{7}$$

The codes are then requantized under the updated (s_c, z_c), and the refined parameters with the smallest value of Eq. equation 5 are selected.

3.3 COARSE-TO-FINE ACTIVATION PRECISION SCHEDULING

3.3.1 MOTIVATION

Uniform activation precision does not account for position-dependent sensitivity in coarse-to-fine autoregressive generation. Early stages contain relatively few tokens but establish the global structural scaffold on which all subsequent stages condition. Later stages operate over denser token grids and mainly refine local content. This ordering gives a practical prior for precision allocation: early perturbations can affect the context consumed by later stages, whereas later perturbations are more localized. CAPS uses this prior by reserving the limited high-precision budget for an early prefix of the autoregressive trajectory.

3.3.2 STAGE-WISE LINEAR FLOPS PROXY

For a quantized linear projection m with input and output dimensions K_m and N_m, processing n tokens requires the linear FLOPs proxy $2nK_mN_m$. At stage i, the scale vector $\mathbf{p}_i$ defines a D-dimensional token grid, and $T_i = \prod_{d=1}^{D} p_{i,d}$ is the number of image tokens in that stage. Thus, image-token projections process BT_i tokens for batch size B. We use F_i as the linear-layer FLOPs proxy for precision allocation. We also include text-conditioning KV projections using the number of conditional tokens T_{cond}. Let $n_{i,m}$ denote the resulting token count associated with projection m at stage i; the stage cost is

$$\begin{aligned} F_i &= \sum_{m \in \mathcal{M}} 2n_{i,m}K_mN_m, \\ n_{i,m} &= \begin{cases} BT_i, & m \text{ is an image-token projection,} \\ T_{\text{cond}}, & m \text{ is a text-KV projection.} \end{cases} \end{aligned} \tag{8}$$

3.3.3 FLOPS-BUDGETED PRECISION ALLOCATION

CAPS assigns an activation bit-width $b_i \in \{b_{\text{low}}, b_{\text{high}}\}$ to each autoregressive stage while enforcing a FLOPs-weighted average-bit constraint:

$$\bar{b}_{\text{act}} = \frac{\sum_{i=1}^{S} F_i b_i}{\sum_{i=1}^{S} F_i} \leq B_{\text{target}}. \tag{9}$$

Equivalently, the FLOPs that can be assigned high precision are bounded by

$$F_{\text{high}}^{\text{budget}} = \left\lfloor \frac{(B_{\text{target}} - b_{\text{low}}) \sum_{i=1}^{S} F_i}{b_{\text{high}} - b_{\text{low}}} \right\rfloor. \tag{10}$$

The choice of b_{low}, b_{high}, and B_{target} specifies the available precision budget. Because stages are indivisible allocation units, the realized $\bar{b}_{\text{act}}$ can be lower than the target.

3.3.4 PREFIX-GREEDY SCHEDULING AND RUNTIME SWITCHING

To instantiate the coarse-to-fine prior, CAPS uses a prefix-greedy allocation. Starting from the earliest stage, it assigns b_{high} until the next stage would exceed Eq. equation 10; all remaining stages use b_{low}. Formally, the high-precision set is a prefix $\mathcal{H} = \{1, \ldots, k\}$, where k is the largest index satisfying

$$\sum_{i=1}^{k} F_i \leq F_{\text{high}}^{\text{budget}}. \tag{11}$$

At inference time, CAPS constructs the plan once from the input scale schedule and switches the activation bit-width of all targeted quantized linear projections at each stage boundary. The prefill computation is initialized at b_{high} to protect the initial conditioning and structural context; Eq. equation 9 is defined over the scheduled autoregressive coarse-to-fine stages. This runtime stage-adaptive behavior uses the same quantized linear modules throughout generation, while their activation precision changes as the generation stage progresses.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models and Evaluation Protocols. We evaluate HGA-Quant on four ARVG backbones: Infinity-2B (Han et al., 2025), HART (Tang et al., 2024), VAR-d30 (Tian et al., 2024), and Flex-VAR (Jiao et al., 2025). For Infinity, our primary testbed, we evaluate text-image alignment using GenEval (Ghosh et al., 2023) and DPG-Bench (Hu et al., 2024), and assess preference-oriented visual quality using ImageReward (Xu et al., 2023). For HART, we evaluate prompt adherence using

Algorithm 1 HGA-Quant pipeline for autoregressive visual generation.

Require: pretrained ARVG model f_θ; target modules $\mathcal{M}$; scale schedule $\mathcal{S}$; quantizer settings $\mathcal{C}$.
Ensure: quantized model $f_{\widehat{\theta}}$ and stage-wise activation plan Π.

$\mathcal{M}_q \leftarrow \varnothing$
apply the fixed random Hadamard representation transform when used by the target implementation
Offline weight calibration:
for each target module $m \in \mathcal{M}$ with weight W_m **do**
    $(Q_m, s_m, z_m) \leftarrow \text{T-BGR}(W_m, \mathcal{C})$
    replace m with a quantized linear module storing (Q_m, s_m, z_m)
    $\mathcal{M}_q \leftarrow \mathcal{M}_q \cup \{m\}$
end for
Runtime activation scheduling:
$\Pi \leftarrow \text{CAPS}(\mathcal{S}, \mathcal{M}_q, \mathcal{C})$
run prefill with high-bit activations
for stage $i = 1, \ldots, |\mathcal{S}|$ **do**
    set all modules in $\mathcal{M}_q$ to activation bit-width Π_i
    generate the tokens of stage i
end for
return $f_{\widehat{\theta}}, \Pi$

DPG-Bench and visual quality using Fréchet Inception Distance (FID) on the MJHQ-30K benchmark (Li et al., 2024a). For VAR-d30 and FlexVAR, we generate 50K images on ImageNet (Deng et al., 2009) and evaluate generation quality using FID (Heusel et al., 2017) and Inception Score (IS) (Salimans et al., 2016). All experiments run on a single NVIDIA RTX A6000 GPU for image generation and evaluation.

Compared Methods. Baselines include RTN (Round-to-Nearest), SmoothQuant (Xiao et al., 2023), SVDQuant (Li et al., 2025a), QuaRot (Ashkboos et al., 2024), and PTQ4ARVG (Liu et al., 2026); PTQ4ARVG uses 128 MS COCO captions for calibration (Lin et al., 2014).

Configurations. We denote b-bit weights and a-bit activations by W_bA_a. *Ours* is uniform-precision T-BGR with $\kappa = 3, \gamma = 8$; *Ours (CAPS)* uses the same weight calibration with FLOPs-budgeted A8/A4 stage scheduling targeting W4A6, whose realized average may fall below 6 bits because stages are indivisible. Our method and QuaRot use fixed Hadamard preprocessing; other baselines use their standard settings. Red/blue mark the best/second-best quantized results.

4.2 Main Results

Infinity. T-BGR is competitive as activation precision decreases: at W4A4 it obtains 0.44772 ImageReward and 80.1773 DPG-Bench, compared with 0.25269 and 75.3323 for QuaRot. Most other methods collapse on at least one metric at W4A4. At the nominal W4A6 target, *Ours (CAPS)* reaches 0.69043, 0.83523, and 82.0877 on GenEval, ImageReward, and DPG-Bench with only 5.5711 FLOPs-weighted activation bits, improving over uniform W4A6 T-BGR (0.68003/0.82882/82.0022).

HART, VAR-d30, and FlexVAR. Uniform T-BGR is strongest at W4A6 on these backbones, while CAPS realizes 5.4963, 5.5000, and 5.5000 effective activation bits, respectively, and loses some quality from the lower final budget. On HART, CAPS reaches 71.058 DPG-Bench and 21.675 FID, versus 73.211 and 13.166 for uniform W4A6; the FID is still better than QuaRot's 23.916, although DPG-Bench is slightly lower than QuaRot's 71.386, and both are far better than the other baselines. On VAR-d30, CAPS obtains 2.9469/283.009 (FID/IS), below uniform W4A6's 2.6586/290.467 but better than QuaRot on both metrics. On FlexVAR, CAPS obtains 8.6066/201.977 versus 2.9150/290.874 for uniform W4A6 and 4.1719/254.551 for QuaRot, while remaining better than other baselines. The reduced activation budget can hurt absolute quality, but coarse-to-fine allocation still outperforms most alternatives at lower compute.

Table 1: Unified comparison on Infinity-2B and HART.

Quantization	Methods	Infinity-2B			HART	
		GenEval ↑	ImageReward ↑	DPG-Bench ↑	DPG-Bench ↑	MJHQ-30K FID ↓
FP	–	0.71896	0.95485	83.4741	74.835	6.178
W8A8	RTN	0.72706	0.90395	83.0017	**75.754**	12.476
	SmoothQuant	0.72192	0.92828	83.0177	75.354	7.185
	SVDQuant	0.71827	0.93168	82.8461	75.411	7.477
	QuaRot	0.72866	0.94160	**83.2944**	75.019	6.380
	PTQ4ARVG	**0.74081**	0.89858	82.971	–	–
	Ours	0.73216	**0.94418**	83.1832	75.172	**6.366**
W4A8	RTN	0.61900	0.33108	79.1849	62.029	92.908
	SmoothQuant	0.51707	-0.14869	75.0448	49.257	117.259
	SVDQuant	0.61737	0.26062	78.9916	**72.656**	26.802
	QuaRot	0.66563	0.75864	80.9206	72.173	16.669
	PTQ4ARVG	0.68797	0.64962	81.6667	–	–
	Ours	**0.69825**	**0.83010**	**82.0691**	71.895	**9.379**
W4A6	RTN	0.44685	-0.29458	71.0519	17.199	186.260
	SmoothQuant	0.28231	-0.93134	66.0039	18.593	145.355
	SVDQuant	0.48178	-0.30246	75.5081	64.644	82.087
	QuaRot	0.65560	0.76999	80.8647	71.386	23.916
	PTQ4ARVG	0.65213	0.50648	81.2235	–	–
	Ours	0.68003	0.82882	82.0022	**73.211**	**13.166**
	Ours (CAPS)	**0.69043**	**0.83523**	**82.0877**	71.058	21.675
W4A4	RTN	0.00312	-2.25143	8.2155	5.340	307.111
	SmoothQuant	0.00141	-2.24901	6.0742	8.749	191.014
	SVDQuant	0.00469	-2.25759	4.8947	10.072	195.629
	QuaRot	0.47687	0.25269	75.3323	50.814	134.535
	PTQ4ARVG	0.02376	-2.18336	26.2649	–	–
	Ours	**0.60207**	**0.44772**	**80.1773**	**61.417**	**88.475**

Table 2: Unified comparison on VAR-d30 and FlexVAR.

Quantization	Methods	VAR-d30		FlexVAR	
		FID ↓	IS ↑	FID ↓	IS ↑
FP	–	1.9827	323.326	2.1598	336.262
W8A8	RTN	1.9742	312.803	2.2093	329.612
	SmoothQuant	2.0138	318.025	2.1797	331.019
	SVDQuant	1.9799	319.311	2.1680	329.660
	QuaRot	1.9708	322.788	**2.1604**	335.229
	PTQ4ARVG	2.4351	314.487	–	–
	Ours	**1.9631**	**323.172**	2.1854	**336.288**
W4A8	RTN	54.7231	45.315	10.4348	157.526
	SmoothQuant	13.1672	146.123	6.3042	209.413
	SVDQuant	6.6533	208.545	5.7541	229.426
	QuaRot	3.9433	255.786	3.6883	264.783
	PTQ4ARVG	26.9862	85.808	–	–
	Ours	**2.5532**	**293.163**	**2.7165**	**297.213**
W4A6	RTN	92.5568	17.685	37.4952	59.187
	SmoothQuant	21.2844	102.013	16.2536	130.247
	SVDQuant	12.4921	150.365	14.8319	146.591
	QuaRot	4.2578	249.087	4.1719	254.551
	PTQ4ARVG	37.2724	63.484	–	–
	Ours	**2.6586**	**290.467**	**2.9150**	**290.874**
	Ours (CAPS)	2.9469	283.009	8.6066	201.977
W4A4	RTN	195.3049	4.439	233.0478	3.004
	SmoothQuant	173.3009	4.683	221.9740	3.719
	SVDQuant	153.7270	5.418	225.9401	3.537
	QuaRot	16.9405	143.519	44.4773	55.159
	PTQ4ARVG	109.2316	11.218	–	–
	Ours	**5.9785**	**232.040**	**17.6975**	**134.295**

CAPS stage sensitivity. Lowering one consecutive stage pair from A8 to A4 reduces ImageReward by 0.17448 for the first pair but only 0.00442 for stages 11–12 (Figure 4). CAPS achieves the lowest FLOPs-weighted activation precision (5.5711 bits) and highest ImageReward (0.83523), supporting a high-precision prefix and low-precision refinement.

4.3 Ablation Study

T-BGR. We investigate the tail-stretch factor γ in Tail-Aware Bounded-init Grid Refinement (T-BGR), which controls the contribution of distribution-tail coefficients when optimizing the scale and zero point. Here, $\gamma = 1$ corresponds to ordinary BGR, while $\gamma > 1$ increases the reconstruction weight of coefficients satisfying $|w - \mu_c| > 3\sigma_c$. We evaluate W4A5.6 with coarse-to-fine mixed precision, uniform W4A6, and uniform W4A4 on Infinity-2B using ImageReward. As shown in Table 3, the main setting $\gamma = 8$ improves over no BGR in all three configurations and reaches 0.83523 for W4A5.6 and 0.82882 for W4A6. The optimum is not monotonic in the tail-stretch factor: W4A4 is highest at $\gamma = 0$, which assigns zero weight to tail reconstruction error and emphasizes the central mass. This suggests that the most aggressive W4A4 regime favors preserving the dense central region over the rare extremes.

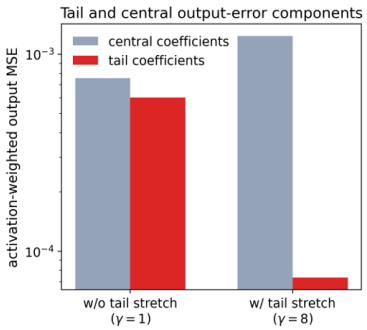


Figure 3: Effect of the tail-stretch factor in T-BGR.

We further isolate the weight-side effect with FP activations and fixed Hadamard preprocessing. Under W4 weight quantization with ordinary BGR ($\gamma = 1$), ImageReward is 0.79494. Restoring the detected tail coefficients to FP raises it to 0.82331 when restoring a random half of the tail entries and to 0.83709 when restoring all of them, indicating that a small tail subset has a disproportionate influence on visual quality. T-BGR then tests whether this insight can be used without an FP outlier branch: assigning a larger tail-stretch factor, $\gamma = 8$, reaches 0.84753, higher than both ordinary BGR and full tail restoration under $\gamma = 1$. This supports treating T-BGR as a range-allocation mechanism that protects tail information while retaining the standard low-bit weight representation.

Table 3: Ablation studies on Infinity-2B. Left: ablation of the T-BGR tail-stretch factor γ. Right: ablation of CAPS. W4A5.6 combines W4A4 and W4A8 at a FLOPs-weighted effective activation precision of 5.5711 bits; "no BGR" omits bounded grid refinement.

	no BGR	$\gamma = 0$	$\gamma = 1$	$\gamma = 2$	$\gamma = 8$	$\gamma = 16$
W4A5.6	0.81594	0.81001	0.81458	0.80069	**0.83523**	0.81491
W4A6	0.82125	0.80062	0.80668	0.79104	**0.82882**	0.79840
W4A4	0.40181	**0.50311**	0.48880	0.49401	0.44772	0.43064

	GenEval↑	**ImageReward↑**	**DPG-Bench↑**
W4A8	0.69825	0.83010	82.0691
W4A6	0.68003	0.82882	82.0022
W4A4	0.60207	0.44772	80.1773
W4A5.6	0.69043	0.83523	82.0877

CAPS. Table 3 evaluates CAPS. W4A5.6 obtains 0.69043, 0.83523, and 82.0877 on GenEval, ImageReward, and DPG-Bench, respectively, compared with 0.68003, 0.82882, and 82.0022 for uniform W4A6. Uniform W4A4 instead drops to 0.60207, 0.44772, and 80.1773. Figure 4 motivates prefix protection; Appendix D compares early-high, late-high, and random schedules.

Activation allocation	Avg. act. bits	**ImageReward ↑**	Δ vs. W4A8
W4A8 (uniform reference)	8.0000	0.83010	0.00000
A4 at stages 1–2	7.9967	0.65562	-0.17448
A4 at stages 3–4	7.9788	0.79154	-0.03856
A4 at stages 5–6	7.9197	0.80950	-0.02060
A4 at stages 7–8	7.7497	0.81915	-0.01095
A4 at stages 9–10	7.3917	0.81671	-0.01339
A4 at stages 11–12	6.5178	0.82568	-0.00442
CAPS: A4 at stages 12–13	**5.5711**	**0.83523**	+0.00513

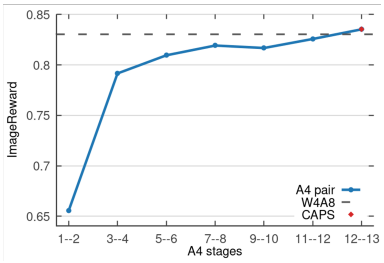


Figure 4: CAPS stage sensitivity on Infinity-2B. The table reports FLOPs-weighted effective activation bits and ImageReward when changing one consecutive stage pair from A8 to A4; the curve plots the same ImageReward values with the uniform W4A8 reference as a dashed horizontal line.

5 Conclusion

We presented HGA-Quant, a training-free PTQ framework for low-bit ARVG that combines tail-aware weight calibration with budgeted activation allocation. Experiments across four backbones demonstrate the benefits of T-BGR under aggressive quantization. CAPS supports protecting early coarse stages, but reduced activation budgets can compromise quality on some backbones. These findings motivate backbone-specific precision schedules to balance generation quality and efficiency.

AI USE STATEMENT

Generative AI tools were used to assist with drafting and modifying experimental code, including initial implementations of core experimental components and auxiliary scripts. They were also used for language polishing, minor revisions to the organization and content of the manuscript, visualization code, and figure layout ideas.

All AI-assisted code was carefully reviewed, tested, and modified before use, and AI-assisted manuscript content was reviewed and verified before inclusion. The core research ideas, methodology, and experimental design were developed independently of generative AI. Experimental results were analyzed and interpreted by the research team, and the conclusions were independently formulated based on these analyses. The authors take full responsibility for the final content of this work, including all text, code, results, and figures.

ETHICS STATEMENT

This work studies post-training quantization methods for improving the efficiency of autoregressive visual generation models. Quantized models may retain the biases, limitations, and potential for misuse of their pretrained counterparts, while reduced computational and memory costs may facilitate broader deployment of such models. Our experiments focus on generation quality and quantization efficiency and do not evaluate the safety, fairness, or potential misuse of the resulting models. The experiments use publicly available models and benchmark datasets and do not involve human subjects or the collection of personal or sensitive data.

REPRODUCIBILITY STATEMENT

We will publicly release the source code and relevant configurations used to obtain the experimental results reported in this paper. The release will cover our quantization implementations for Infinity, HART, VAR, and FlexVAR, together with the corresponding experimental configurations, to facilitate reproduction of our results.

REFERENCES

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. *arXiv preprint arXiv:2404.00456*, 2024. URL https://arxiv.org/abs/2404.00456.

Zigeng Chen, Xinyin Ma, Gongfan Fang, and Xinchao Wang. Collaborative decoding makes visual auto-regressive modeling efficient. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, 2025. URL https://arxiv.org/abs/2411.17787.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, 2009.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2310.11513.

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 *Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2412.04431.

Yefei He, Feng Chen, Yuanyu He, Shaoxuan He, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. Zipar: Parallel auto-regressive image generation through spatial locality. *arXiv preprint arXiv:2412.04062*, 2024. URL https://arxiv.org/abs/2412.04062.

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

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment, 2024.

Siyu Jiao, Gengwei Zhang, Yinlong Qian, Jiancheng Huang, Yao Zhao, Humphrey Shi, Lin Ma, Yunchao Wei, and Zequn Jie. Flexvar: Flexible visual autoregressive modeling without residual prediction. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, 2025. URL https://arxiv.org/abs/2502.20313.

Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2.5: Three insights towards enhancing aesthetic quality in text-to-image generation, 2024a. URL https://arxiv.org/abs/2402.17245.

Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://arxiv.org/abs/2411.05007.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 2024b.

Zhiteng Li, Hanxuan Li, Junyi Wu, Kai Liu, Qin Haotong, Linghe Kong, Guihai Chen, Yulun Zhang, and Xiaokang Yang. Dvd-quant: Data-free video diffusion transformers quantization. *arXiv preprint arXiv:2505.18663*, 2025b.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for llm compression and acceleration. In *MLSys*, 2024.

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*, 2014. URL https://arxiv.org/abs/1405.0312.

Jinlai Liu, Jian Han, Bin Yan, Hui Wu, Fengda Zhu, Xing Wang, Yi Jiang, Bingyue Peng, and Zehuan Yuan. Infinitystar: Unified spacetime autoregressive modeling for visual generation. In *Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2511.04675.

Xuewen Liu, Zhikai Li, Jing Zhang, Mengjuan Chen, and Qingyi Gu. Ptq4arvg: Post-training quantization for autoregressive visual generation models. *International Conference on Learning Representations*, 2026.

Jaehyeon Moon and Bumsub Ham. Shift-and-sum quantization for visual autoregressive models. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2606.16131.

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Neural Information Processing Systems*, 2016. URL https://arxiv.org/abs/1606.03498.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang Zhang, Peng Gao, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. *arXiv preprint arXiv:2308.13137*, 2023.

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. *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2410.10812.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2404.02905.

Enrico Vezzali, Federico Bolelli, Costantino Grana, Luca Benini, and Yawei Li. Enabling 8b bitwise autoregressive image generation on edge gpus. In *Pattern Recognition - 28th International Conference, ICPR 2026, Lyon, France, August 17-22, 2026, Proceedings, Part XII*, 2026.

Jiahuan Wang, Yuxin Chen, Jun Yu, Guangming Lu, and Wenjie Pei. Editinfinity: Image editing with binary-quantized generative models. *Advances in Neural Information Processing Systems*, 2025.

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.

Renjie Wei, Songqiang Xu, Qingyu Guo, and Meng Li. Fpqvar: Floating point quantization for visual autoregressive model with fpga hardware co-design. *arXiv preprint arXiv:2505.16335*, 2025.

Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. Outlier suppression+: Accurate quantization of large language models by equivalent and optimal shifting and scaling. *arXiv preprint arXiv:2304.09145*, 2023.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. URL https://arxiv.org/abs/2211.10438.

Rui Xie, Tianchen Zhao, Zhihang Yuan, Rui Wan, Wenxi Gao, Zhenhua Zhu, Xuefei Ning, and Yu Wang. Litevar: Compressing visual autoregressive modelling with efficient attention and quantization. *arXiv preprint arXiv:2411.17178*, 2024.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: learning and evaluating human preferences for text-to-image generation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2304.05977.

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

Xun Zhang, Kaicheng Yang, Hongliang Lu, Haotong Qin, Yong Guo, and Yulun Zhang. Q-dit4sr: Exploration of detail-preserving diffusion transformer quantization for real-world image super-resolution. *arXiv preprint arXiv:2602.01273*, 2026.

A ALGORITHM

Algorithm 2 Tail-Aware Bounded-init Grid Refinement (T-BGR).

Require: weight matrix W; bit-width b_w; grid size N; tail threshold κ; tail weight γ; tolerance ϵ; iteration limit $I_{\max}$.
Ensure: integer codes $Q^\star$; real-valued scales $s^\star$; integer zero points $z^\star$.

$q_{\max} \leftarrow 2^{b_w} - 1$
compute per-channel statistics μ_c, σ_c of $W_{c,:}$
$\rho_{c,j} \leftarrow 1 + (\gamma - 1)\mathbf{1}(|w_{c,j} - \mu_c| > \kappa\sigma_c)$
initialize channel-wise best loss $\mathcal{L}_c^\star \leftarrow +\infty$
$\ell_c \leftarrow \min_j w_{c,j}$, $u_c \leftarrow \max_j w_{c,j}$, and $\eta_c \leftarrow (u_c - \ell_c)/(2N)$
for $r = 0, \ldots, N-1$ **do**
    $W_{c,:}^{(r)} \leftarrow \mathrm{clip}(W_{c,:}, \ell_c + r\eta_c, u_c - r\eta_c)$ ▷ bounded initialization
    initialize (s, z, Q) from the asymmetric affine quantizer induced by $W^{(r)}$
    for $t = 1, \ldots, I_{\max}$ **do**
        $A \leftarrow Q - z$
        $s_c \leftarrow \dfrac{\sum_j \rho_{c,j} A_{c,j} w_{c,j}}{\sum_j \rho_{c,j} A_{c,j}^2}$
        $z_c \leftarrow \mathrm{clip}\left(\mathrm{round}\left(\dfrac{\sum_j \rho_{c,j}(q_{c,j} - w_{c,j}/s_c)}{\sum_j \rho_{c,j}}\right), 0, q_{\max}\right)$
        $Q \leftarrow \mathrm{clip}(\mathrm{round}(W/s) + z, 0, q_{\max})$
        $\mathcal{L}_c \leftarrow \dfrac{\sum_j \rho_{c,j}\left(w_{c,j} - (q_{c,j} - z_c)s_c\right)^2}{\sum_j \rho_{c,j}}$
        if $\max_c |\mathcal{L}_c - \mathcal{L}_c^{\mathrm{prev}}| < \epsilon$ **then**
            break
        end if
        $\mathcal{L}^{\mathrm{prev}} \leftarrow \mathcal{L}$
    end for
    update $(Q_{c,:}^\star, s_c^\star, z_c^\star, \mathcal{L}_c^\star)$ for channels with $\mathcal{L}_c < \mathcal{L}_c^\star$
end for
return $Q^\star, s^\star, z^\star$

Algorithm 3 Coarse-to-Fine Activation Precision Scheduling (CAPS).

Require: scale schedule $\mathcal{S} = \{\mathbf{p}_i\}_{i=1}^S$; quantized modules $\mathcal{M}_q$; effective batch size B; conditional token count T_{cond}; bit-widths $b_{\mathrm{low}} < b_{\mathrm{high}}$; target average bit-width b_{tar}.
Ensure: stage-wise activation plan $\Pi = (\Pi_1, \ldots, \Pi_S)$.

for $i = 1, \ldots, S$ **do**
    $T_i \leftarrow \prod_d p_{i,d}$
    $F_i \leftarrow \sum_{m \in \mathcal{M}_q} 2\, n_{i,m}\, d_m^{\mathrm{in}} d_m^{\mathrm{out}}$, where $n_{i,m} = T_{\mathrm{cond}}$ for text-KV projections and $n_{i,m} = BT_i$ otherwise
end for
$F_{\mathrm{budget}} \leftarrow \left\lfloor \dfrac{b_{\mathrm{tar}} - b_{\mathrm{low}}}{b_{\mathrm{high}} - b_{\mathrm{low}}} \sum_{i=1}^S F_i \right\rfloor$
choose the longest prefix $\mathcal{H} = \{1, \ldots, k\}$ such that $\sum_{i \in \mathcal{H}} F_i \le F_{\mathrm{budget}}$
for $i = 1, \ldots, S$ **do**
    $\Pi_i \leftarrow b_{\mathrm{high}}$ if $i \in \mathcal{H}$, otherwise $\Pi_i \leftarrow b_{\mathrm{low}}$
end for
return Π

B WEIGHT-ONLY OUTLIER DIAGNOSTIC UNDER W3

The main ablation uses W4 to study tail-aware weight calibration. To check whether the same intuition remains relevant under a more aggressive weight-only setting, we also run W3 quantization with FP activations and fixed Hadamard preprocessing. Without BGR, W3 collapses to -2.25053 ImageReward, while ordinary BGR ($\gamma = 1$) recovers the score to 0.58618. This confirms that a range-refinement baseline is important before studying finer tail behavior in very low-bit weights.

Starting from W3 with ordinary BGR, restoring all detected tail coefficients to FP further improves ImageReward from 0.58618 to 0.61887. Restoring only a random half gives 0.57379, slightly below the non-restored run, suggesting that the W3 regime is sensitive to uneven mixed-precision perturbations rather than following a simple monotonic rule. The full-restoration result nevertheless supports the same qualitative conclusion as W4: tail precision remains useful when the bit-width is pushed lower.

We also evaluate the stronger T-BGR setting $\gamma = 8$ under W3. It obtains 0.32632, far above the no-BGR collapse but below ordinary BGR. This does not negate the tail-aware motivation; instead, it shows that the appropriate tail emphasis depends on the available bit-width. In an extremely narrow W3 codebook, aggressively stretching the tail can overly reduce the resolution of the central mass, so a bit-width-adaptive tail factor may be preferable.

C COARSE-TO-FINE GENERATION VISUALIZATION

Figure 5 provides an intuitive view of the coarse-to-fine generation process of Infinity. The early stages mainly establish global structures and compositions, while later stages progressively refine object details and visual appearance. This observation motivates stage-aware precision allocation: higher precision can be preserved for early structural generation, while lower precision can be applied to later refinement stages.

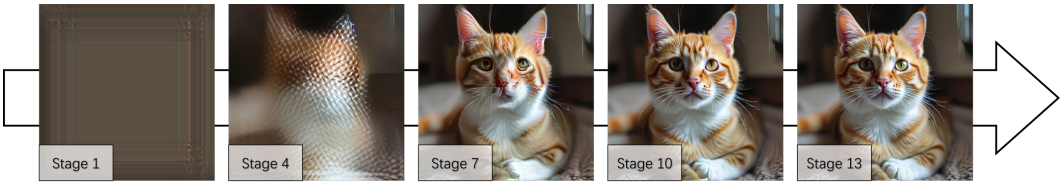


Figure 5: Compact illustration of coarse-to-fine generation at selected stages. The generation process gradually evolves from coarse layouts to detailed images, with early stages determining global structures and later stages refining object-level details.

To further illustrate this behavior, Figure 6 compares intermediate outputs at all 13 stages under full precision (FP), uniform W4A6, CAPS with W4A5.6, and uniform W4A4. The early stages preserve the global layout, while later stages mainly contribute incremental refinements. CAPS maintains a coherent trajectory after switching to lower precision in the final stages, whereas uniform W4A4 introduces artifacts from early stages and produces a visibly degraded trajectory.

D BUDGETED ALLOCATION COMPARISON

The stage-wise results in the main text establish that early and late stages have different precision sensitivity. We therefore add a broader placement check under the same target FLOPs budget. The early-prefix schedule is the CAPS allocation used by our method, with A8 assigned to stages 1–11 and A4 assigned to stages 12–13. The late-high schedule assigns A8 only to the final refinement stage, which has the largest token grid. For random schedules, we start from all-A4 stages, sample a deterministic random stage order, and switch a stage to A8 when doing so does not exceed the target budget; if a stage would exceed the budget, it is skipped and the remaining stages are considered. All rows use the common allocation-comparison entry; consequently, its early-high score is 0.83523.

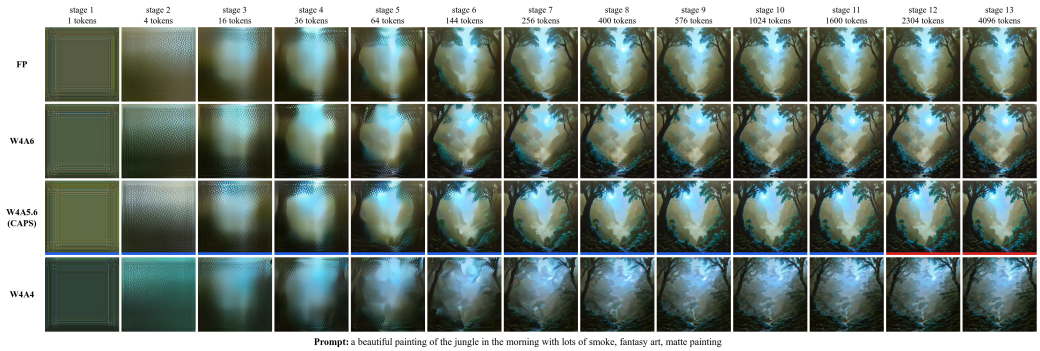


Figure 6: Stage-wise visualization of coarse-to-fine generation. We compare full precision (FP), uniform W4A6, CAPS with W4A5.6, and uniform W4A4 over all 13 generation stages. CAPS preserves the coarse structure with its high-precision prefix and maintains coherent refinement after switching to lower precision in the final stages, whereas uniform W4A4 exhibits visible artifacts throughout the trajectory.

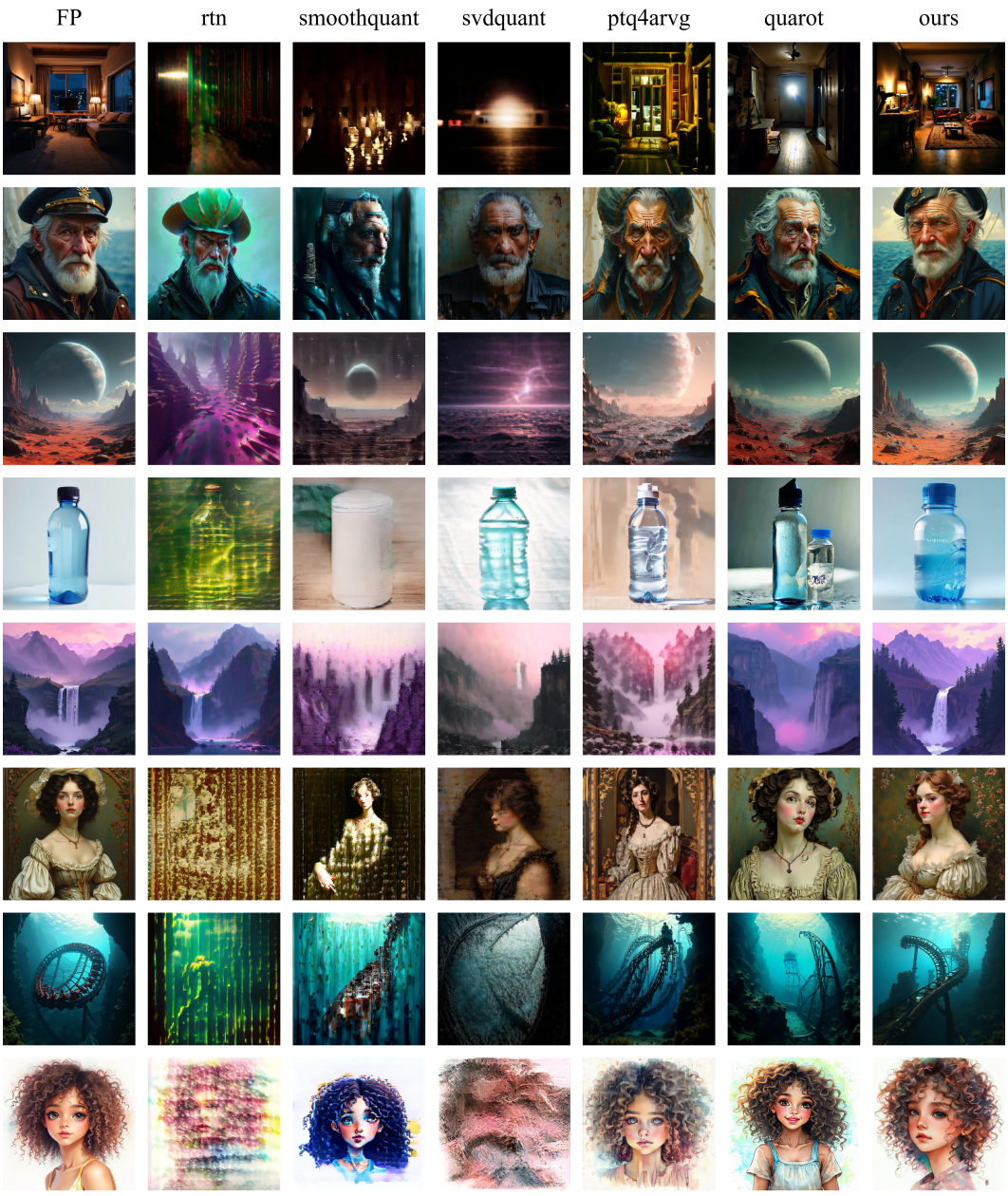
Table 4: Budgeted CAPS allocation comparison on Infinity-2B. Stage labels are one-based. The reported average bit-width is the FLOPs-weighted effective activation precision realized by each allocation, and all rows use the common allocation-comparison evaluation entry.

Allocation	A8 stages	Avg. act. bits	**ImageReward** ↑	Δ vs. early-high
early-high (CAPS)	1–11	5.5711	**0.83523**	0.00000
late-high	13	5.5543	0.44577	-0.38946
`random_01`	1,2,3,4,5,6,8,9,11,12	5.9587	0.81186	-0.02337
`random_02`	1,2,3,4,5,10,13	5.9930	0.77997	-0.05526
`random_03`	1,2,3,4,5,6,7,8,9,10,12	5.8381	0.82448	-0.01075
`random_04`	1,2,3,4,5,6,7,8,13	5.9095	0.80030	-0.03493
`random_05`	1,2,3,4,5,8,9,13	5.9755	0.78069	-0.05454

The early-high and late-high schedules are also close in compute: their high-precision linear FLOPs are 31.20T and 30.87T, differing by 1.06% of the high-precision portions and only 0.42% of the full stage FLOPs. Nevertheless, ImageReward falls from 0.83523 to 0.44577, a decrease of 0.38946 that is close to uniform W4A4 (0.44772). Among the random schedules, the best result is `random_03` at 0.82448, still 0.01075 below early-high despite using a higher realized activation precision of 5.8381 bits. Thus, neither a similar compute budget nor a larger average precision is sufficient without protecting the early stages.

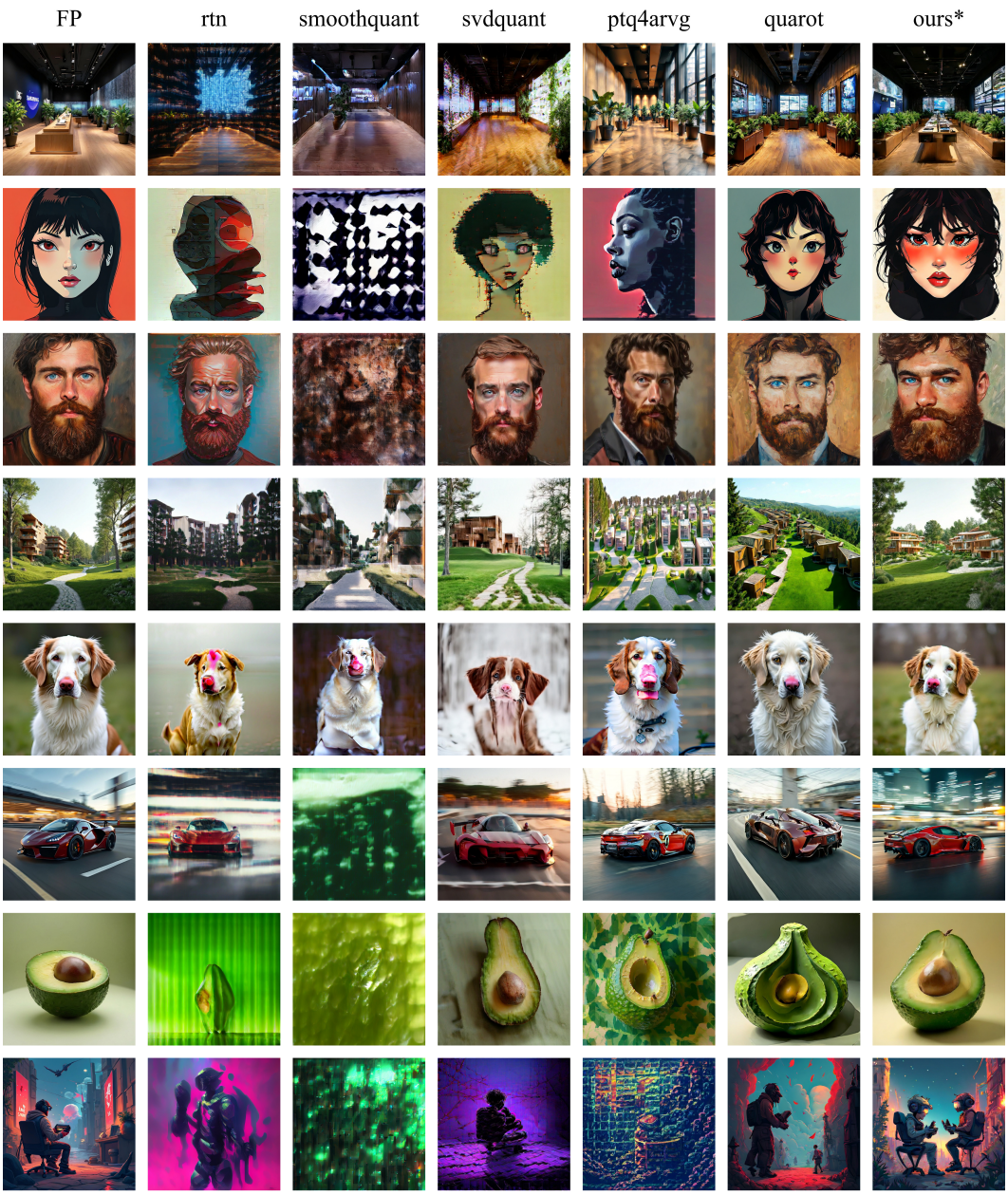
E Qualitative Comparisons

We provide qualitative comparisons across the evaluated methods under W4A6 and coarse-to-fine W4A5.6 quantization on Infinity-2B. For each example, all methods are conditioned on the same text prompt. Our method generally preserves the global composition and fine-grained details more faithfully, while several baselines exhibit visible artifacts, structural distortion, or severe quality degradation at aggressive bit-widths.



Prompt 1: photo of a interior taken with a cheap digital camera at night flash lighting
Prompt 2: Portrait of an old sea captain, male, detailed face, fantasy, highly detailed, cinematic, art painting by greg rutkowski
Prompt 3: surface of an alien planet
Prompt 4: water bottle
Prompt 5: mountains range with waterfall, purple haze, art by greg rutkowski and magali villeneuve
Prompt 6: victorian lady, painting by rossetti, daniel gerhartz, alphonse mucha, bouguereau, detailed art
Prompt 7: an underwater rollercoaster, cinematic, dramatic, -
Prompt 8: beautiful girl with curly hair, watercolor style, colorful, pastel colors, ultra detailed

Figure 7: Comparisons across different methods. In this figure, all images were obtained with the model undergoing W4A6 quantization.



Prompt 1: hyperdetailed samsung store, oak parquet, black walls, digital walls, plants, light corona octane
Prompt 2: face icon stylized minimalist trigun, loftis, cory behance hd by jesper ejsing, by rhads, makoto shinkai and lois van baarle, ilya kuvshinov, rossdraws global illumination
Prompt 3: a portrait oil painting of a smart man in his mid - 4 0 s, brown hair, blue eyes, salt and pepper beard, full beard, red skin, big eye brows, highly detailed, artistic, intricate
Prompt 4: natural suburb with multiple low rise apartment buildings in a park like setting with green hilly lawns and lush trees, pine wooden walls, rustic, large glass windows, cobblestone, grass, white, natural pathways, natural materials, minimalist, swedish design, bright, feng shui, modern, technology, frank lloyd wright
Prompt 5: a nova scotia duck tolling retriever with white chest and pink nose, professional, photography, of, very
Prompt 6: a sports car, motion blur
Prompt 7: photograph of table shaped like an avocado, product, intricate detail
Prompt 8: video game, art

Figure 8: In this figure, the images labeled ”ours*” were obtained using our coarse-to-fine W4A5.6 approach, while the images from other methods were generated using W4A6 quantization.