Enabling 8B Bitwise Autoregressive Image Generation on Edge GPUs

Enrico Vezzali[1,3], Federico Bolelli[1]✉, Costantino Grana[1], Luca Benini[2], and Yawei Li[2]

[1] AImageLab, University of Modena and Reggio Emilia, Italy
{name.surname}@unimore.it
[2] Integrated Systems Laboratory, ETH Zürich, Switzerland
{lbenini, yawli}@iis.ee.ethz.ch
[3] Datalogic, S.p.A., Bologna, Italy

Abstract. Visual Autoregressive (VAR) models face a severe "Memory Wall" on edge devices due to large model size and substantial KV-cache requirements. In this work, we analyze the Infinity VAR family (2B and 8B) and propose a compression pipeline for deployment on constrained NVIDIA Jetson systems. We diagnose critical bottlenecks: activation outliers reaching 353× the median and channel-skewed cache variance. To address this, we propose a hybrid pipeline combining SVDQuant to structurally decouple weight outliers and asymmetric per-channel KV8 quantization. Our approach reduces the Infinity-8B footprint by 64% (37.1GB → 13.3GB), fitting it on the mid-range Orin NX with a 4.1× speedup over Flux.1-dev (W4A4), while achieving superior aesthetic alignment (ImageReward 1.13 vs 0.935). Crucially, we also unlock entry-level feasibility for the Infinity-2B, compressing it from 16.0GB to 7.71GB to enable deployment on the Orin Nano. These results establish a new efficiency standard for high-fidelity generative AI at the edge. The code is available at https://github.com/Henvezz95/VAR-Compressor.

Keywords: Text-to-Image Generation, Quantization, VAR

1 Introduction

Visual Autoregressive (VAR) models [9,22,23] are rapidly emerging as formidable competitors to Diffusion Models in the field of generative computer vision. By adapting LLM scaling laws to "next-scale prediction," architectures like **Infinity** [9] achieve state-of-the-art fidelity with models scaling up to 8 billion parameters. However, the model size growth dictated by the scaling rules introduces a critical bottleneck: the *Memory Wall*. Unlike diffusion models, whose memory footprint remains constant during inference, VAR requires an additional Key-Value (KV) cache that grows with sequence length. The high memory demands effectively lock these systems behind data-center hardware, creating a

✉ Corresponding author: federico.bolelli@unimore.it

barrier for Edge AI applications requiring high-fidelity generation. A primary application is Robotics, where text-to-image models are evolving into "real-time visual planners"; for instance, the DALL-E-Bot framework utilizes generative models to synthesize goal states to guide visual servoing loops [6]. Similarly, in bandwidth-denied environments like search-and-rescue, generative models enable Semantic Compression [18], reconstructing video from tokens—a workflow impossible without a local decoder. Finally, for Consumer Electronics, local execution is mandated by privacy, as cloud execution exposes sensitive user inputs (e.g., biometrics) to model inversion attacks [8].

While model quantization is established for LLMs, its application to VAR is nascent. Attempts like FPQVAR [30] often rely on custom FPGAs or exotic formats, failing to address the commodity GPU standard (INT4/INT8) required for mass adoption. Others, like LiteVAR [33], achieve GPU compatibility but fail to quantize both weights and activations to 4 bits without destroying quality.

In this work, we present a holistic quantization framework to democratize billion-scale VAR models. We adopt a multi-tier hardware strategy: targeting the NVIDIA Jetson Orin Nano (8GB) for the 2B variant, while enabling the flagship 8B model on mid-range Orin NX (16GB). By demonstrating that state-of-the-art generative AI functions on these "extreme edge" devices, we challenge the assumption that high-fidelity generation requires cloud-grade infrastructure. Our contributions are threefold:

1. System-Level Analysis. We conduct the first rigorous sensitivity analysis of the Infinity architecture, revealing that, like LLMs, VAR models exhibit "massive activation" outliers that degrade standard INT4 quantization.

2. Hybrid Quantization Pipeline. We propose a novel integration of **SVDQuant** [15] (for W4A4 weight-activation compression) with a tailored **Asymmetric Per-channel** KV-cache INT8 quantization. This strategy reduces the total system footprint by up to 64%, fitting the flagship 8B model comfortably within the 16GB envelope of mid-range edge devices.

3. Industrial Validation. We measure peak memory use, inference speed, and generation quality on an NVIDIA Jetson platform. In particular, our quantized Infinity 8B model achieves superior ImageReward (1.13 vs 0.935 on MJHQ) and CLIP Score (27.0 vs 25.8 on MJHQ) compared to Flux.1-dev (SVDQuant W4A4), while being 4.1× faster (27s vs 112s). We also open-sourced our code and weights to facilitate reproducibility.

To our knowledge, this is the first framework to successfully apply 4-bit weights and INT8 KV-caching to visual autoregressive models on standard GPU hardware, bridging the gap between scaling laws and edge feasibility.

2 Related Work

2.1 Visual Autoregressive Modeling

Autoregressive (AR) models revolutionized image generation by adapting the "next-token prediction" paradigm of LLMs to the visual domain [12, 21, 35]. To

make this computationally feasible, VQ-based methods such as VQ-VAE [25] and VQ-GAN [5] employ vector quantization to compress images into a sequence of discrete indices from a learned codebook. A decoder-only transformer then predicts these indices in a raster-scan order.

However, this raster-scan constraint imposes a structural bottleneck: generating a sequence of length N requires N serial steps. This strictly sequential dependency precludes parallelization and incurs a quadratic computational cost, significantly limiting global structural planning during early generation. Additionally, standard VQ-VAE approaches are bounded by the information loss inherent in discrete quantization, often resulting in reconstructed images that lack fine-grained fidelity.

To address the serialization issue, visual autoregressive modeling [23] redefines the objective as "next-scale prediction." Instead of flattening the image into a 1D sequence, VAR conceptualizes it as a hierarchy of multi-scale maps. The generation proceeds coarse-to-fine: the model predicts a low-resolution token map and progressively generates higher-resolution maps conditioned on previous scales. While domain-specific super-resolution algorithms can achieve extremely fast, multi-step upsampling directly on edge CPUs [27], generative "next-scale" approaches utilize deep parameter redundancies to hallucinate complex structural consistencies that standard spatial interpolation cannot match.

Recent works address the VQ-VAE information loss issue, enabling the scaling of this paradigm to compete with diffusion models. HART [22] integrates continuous diffusion to refine residuals, while Infinity [9] replaces discrete codebooks with a *Bitwise Tokenizer* and Infinite-Vocabulary Classifier (IVC). By predicting binary vectors, Infinity achieves photorealistic generation and unlocks the potential of billion-parameter models (up to 8B). However, this capacity introduces a "Memory Wall" that renders deployment on edge hardware prohibitive.

2.2 Quantization of Image Generation Models

Post-Training Quantization (PTQ) is well-established for diffusion models. Techniques like SmoothQuant [31] enable 8-bit weight and activation (W8A8) inference by migrating activation outliers to weights, while SVDQuant [15] pushes compression to 4-bit (W4A4) via low-rank decomposition. However, quantization for VAR models is nascent and often hardware-specific. LiteVAR [33] proposes efficient attention pruning and achieves W8A8 quantization, but reports significant degradation at lower bit-widths (W4A8) due to sensitive bottleneck layers. FPQVAR [30] achieves 4-bit quantization by employing custom floating-point formats (FP4) and "Group-wise Hadamard Transformations" to manage asymmetric distributions. Yet, FPQVAR relies on FPGA co-design to execute these non-standard formats, rendering it incompatible with standard industrial edge GPUs where INT4 is the primary low-precision format.

Furthermore, purely weight-centric methods common in LLMs, such as GPTQ [7] or AWQ [16] (W4A16), leave activation memory uncompressed. This creates a memory bandwidth bottleneck that severely throttles inference speed on embedded platforms.

In contrast, we focus on W4A4 quantization for commodity edge hardware. We identify that Infinity's 8B architecture exhibits the "Massive Activation" phenomenon [4] typical of large-scale Transformers. We are the first to adapt the SVDQuant paradigm to VAR models, demonstrating that its outlier-aware decomposition is the critical enabler for preventing quality collapse while running efficiently on standard integer tensor cores.

2.3 Caching Long-Context Generation

For high-resolution generation ($1\,024 \times 1\,024$), the KV-cache can even exceed the memory footprint of the model's weights. While recent methods like HACK [20] and ScaleKV [14] employ dynamic token pruning, they introduce control-flow overheads that can destabilize embedded latency. We prioritize deployment stability and propose a simple but effective **Asymmetric Per-channel** INT8 Quantization alongside weight and activation compression. Crucially, our approach diverges from standard LLM cache quantization methods like KIVI [17], which typically quantize Values on a per-token basis to handle token-wise outliers. In contrast, we observe that for visual autoregressive models, activation variance is predominantly **channel-driven** across both Keys and Values. Leveraging this insight, we apply per-channel asymmetric INT8 quantization uniformly. This deterministic strategy reduces the cache footprint by 50% without the runtime overhead of pruning, providing a robust, architecturally aligned baseline for industrial deployment.

3 Methodology

3.1 Problem Formulation

The deployment of VAR models is constrained by two distinct memory pressures: the static footprint of the model parameters and the dynamic footprint of the Key-Value (KV) cache. While the transformer weights are fixed, the generation process requires holding activation buffers, the VAE decoder, and the text encoder in memory. Furthermore, the KV-cache grows monotonically during generation. For the Infinity 8B model generating a $1\,024 \times 1\,024$ image, the combined load of parameters, cache, and auxiliary encoders (VAE, Text) results in a peak system memory demand of 37.1GB. This magnitude strictly confines high-fidelity generation to data-center-class GPUs. Our objective is to break this "Memory Wall" by compressing the dominant components to fit within commodity edge-device constraints. Specifically, we target 4-bit weight quantization to minimize the static load and 8-bit cache compression, enabling the 8B model to run on mid-range devices (e.g., Orin NX 16GB) and the 2B model on entry-level systems (e.g., Orin Nano 8GB).

3.2 Model Analysis: the Anatomy of Outliers

To achieve the effective quantization goals outlined in Sec. 3.1, we first diagnose the distributional properties of the Infinity models. We focus on two critical

Table 1. Activation Outlier Analysis. We profile the Layer-wise Absolute Max-to-Median Ratio and Kurtosis on 12K calibration tokens. For each layer, we compute these metrics on the flattened activation volume (pooling tokens and channels). We then report the mean and maximum values observed across all layers of the same type.

Layer Type	Infinity 2B				Infinity 8B			
	Max/Median		Kurtosis		Max/Median		Kurtosis	
	Mean	Max	Mean	Max	Mean	Max	Mean	Max
QKV Projection	30.6	47.2	5.55	10.89	37.2	59.4	6.59	21.97
Out Projection	23.8	128.0	2.72	24.67	27.6	195.4	3.40	23.90
FFN Up-Proj	24.4	36.5	2.99	7.99	24.0	46.6	2.10	12.24
FFN Down-Proj	**98.2**	**353.0**	**26.41**	**153.80**	**63.4**	**226.3**	**26.13**	**113.50**

Table 2. KV-Cache Distributional Statistics. Analysis of coefficient of variation (CV) and skewness (γ) on 12K tokens. The dominance of CV_{chan} over CV_{tok} confirms variance is channel-driven.

Cache Type	Model	Coefficient of Variation		Skewness (γ)	
		CV Channel	CV Token	Mean	Max
Key Cache	2B	**0.28**	0.07	0.20	11.56
	8B	**0.31**	0.06	0.17	7.67
Value Cache	2B	**0.25**	0.17	0.11	4.43
	8B	**0.20**	0.17	0.11	6.67

bottlenecks: the activation statistics in linear layers and the variance structure of the KV-cache.

Activation Statistics. Large Transformer-based models often exhibit the "Massive Activation" phenomenon [4], where the distribution of layer activations contains extreme outliers. To determine if this persists in the Infinity architecture, we profiled activations using 12K random tokens from the step-aware calibration set described in Sec. 4.1. For each layer, we reshaped the activation tensor to a 2D structure (tokens $\times$ channels) and computed the Kurtosis and the Absolute Max-to-Median Ratio ($\eta = \max(|X|)/\text{median}(|X|)$). These metrics serve as standard indicators of the "tailedness" of a distribution. We report the mean and maximum of these metrics across all layers of the same type in Table 1.

The results indicate that Infinity is highly sensitive to outliers. All profiled layers maintain a mean Max-to-Median ratio above 20×. Crucially, the FFN Down-Projection emerges as the structural bottleneck, exhibiting the most severe outliers for both the 2B and 8B models. As illustrated in Fig. 1 (left), the Infinity-2B model is particularly "sharp", with outlier peaks reaching 353× the median signal. This heightened severity in the smaller model suggests it is structurally more difficult to quantize than its 8B counterpart.

Cache Diagnostic Metrics. To characterize the KV-cache, we analyze the Dynamic Range $R = \max(\mathbf{x}) - \min(\mathbf{x})$ rather than raw magnitudes, as R directly determines the resolution of the quantization grid. We define the Coefficient of Variation ($CV = \sigma/\mu$) of R across channels and tokens to quantify the relative dispersion of the dynamic range. We denote the coefficient of variation across channels as CV_{chan} and across tokens as CV_{tok}.

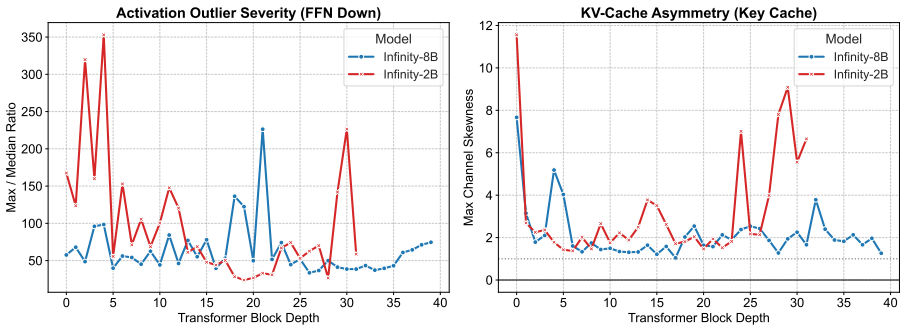

Fig. 1. Left: Activation outlier severity (highest per-channel Absolute Max/Median Ratio) in the FFN Down-Projection layers across network depth. **Right**: Maximum channel skewness in the Key Cache layers. While average skewness is low, specific blocks show significant asymmetry.

Additionally, we compute the Fisher-Pearson skewness (third standardized moment) γ per channel, reporting the mean absolute value $|\bar{\gamma}|$ and the maximum $|\gamma_{max}|$ (across all channels and layers) to assess symmetry.

Diagnosis: Channel-Driven Variance & Skew. Our analysis, detailed in Table 2, reveals key insights into the variance of the KV-cache.

First, variance is strictly **channel-dependent**. In the Infinity 8B Key cache, the channel-wise variation ($CV_{chan} \approx 0.31$) is over $5\times$ higher than the token-wise variation ($CV_{tok} \approx 0.06$). We observe the same dominance in the 2B model. The value cache also exhibits $CV_{chan} > CV_{tok}$. This confirms that outlier ranges are static properties of specific embedding dimensions rather than transient artifacts of input tokens, justifying the use of static per-channel scaling.

Second, we observe significant distributional asymmetry in some layers. While the average skewness remains moderate, specific channels exhibit extreme skew. As shown in Fig. 1 (right), both the 2B and 8B models contain channels with skewness $|\gamma_{max}| > 7$ (peaking at 11.56 for the 2B Key Cache). This implies the dynamic range is heavily biased away from zero. Consequently, standard symmetric quantization schemes (e.g., $[-127, +127]$) would be inefficient, wasting bit-width on empty ranges. We therefore adopt per-channel asymmetric quantization to align scaling factors with the axes of highest variance and shift zero-points to accommodate the skew.

3.3 Weight & Activation Quantization Strategy

To enable 4-bit inference, we must identify a quantization scheme capable of handling the extreme outliers identified in Sec. 3.2. We evaluate three progressively sophisticated strategies.

1. Baseline: Block-Wise Quantization. We establish a standard baseline using W4A4 Dynamic Block-Wise quantization, a common standard for low-bit

LLM quantization. Unlike standard per-channel (weights) or per-token (activations) granularity, block-wise quantization partitions the channel dimension into fixed-size groups ($B = 64$ for our baseline) to isolate outliers. For a local block vector $\mathbf{x} \in \mathbb{R}^B$, the dynamic scale s_b is computed as:

$$s_b = \frac{\max(|\mathbf{x}|)}{2^{b-1} - 1}, \quad \mathbf{x}_{int} = \text{clamp}\left(\left\lfloor \frac{\mathbf{x}}{s_b} \right\rceil, -8, 7 \right). \tag{1}$$

Failure Analysis: Despite this fine granularity, the Infinity-2B model suffers a catastrophic collapse (ImageReward 0.981 $\rightarrow$ 0.616 on MJHQ, shown in Table 5). As diagnosed in Table 1, the FFN outliers (353×) are so severe that they dominate even small blocks of 64 parameters, destroying the signal of the non-outlier weights.

2. Activation Smoothing (SmoothQuant). To mitigate this, we evaluated **SmoothQuant** [31], which mathematically migrates the quantization difficulty from activations to weights. The key insight is that activation outliers persist in specific channels across all tokens. By introducing a per-channel smoothing factor $\mathbf{s} \in \mathbb{R}^{C_{in}}$, we can mathematically migrate the quantization difficulty from the activations to the weights. For a linear layer $Y = XW$, the operation is transformed as $Y = (X \cdot \text{diag}(\mathbf{s})^{-1}) \cdot (\text{diag}(\mathbf{s}) \cdot W) = \hat{X} \cdot \hat{W}$.

While SmoothQuant is often effective for large models (e.g., LLaMA-65B), our results show it is insufficient for the Infinity-2B architecture. As shown in Table 5, SmoothQuant only marginally improves the ImageReward (0.616 $\rightarrow$ 0.667 on MJHQ), failing to recover the FP16 fidelity (0.981).

3. The Solution: Outlier-Aware Decomposition (SVDQuant). Consequently, we adopt **SVDQuant** [15], which does not merely smooth outliers but *structurally decouples* them. This method decomposes the weight matrix into a dense low-precision component and a sparse high-precision component:

$$\hat{W} \approx \hat{W}_{int4} + L_1 L_2. \tag{2}$$

Here, $\hat{W}_{int4}$ captures the bulk signal, while the low-rank term $L_1 L_2$ (rank $r = 32$) explicitly preserves the high-magnitude outlier directions in FP16. This strategy provides the breakthrough required for the 2B model, boosting ImageReward from 0.667 (SmoothQuant) to 0.848 (SVDQuant), effectively recovering most of the model's generative capabilities.

Inference Integration. We leverage the **Nunchaku** engine [15] to execute this dual-branch operation efficiently. The low-rank down-projection ($X \cdot L_1$) is fused with activation quantization, and the up-projection is fused with the 4-bit GEMM kernel. To deploy this within the Infinity architecture, we developed a custom PyTorch wrapper[4] that exposes Nunchaku's optimized C++/CUDA kernels as standard nn.Linear modules. This integration allows us to seamlessly replace Infinity's linear layers with SVDQuant-optimized counterparts while maintaining the flexibility of the original Python-based autoregressive loop.

[4] Available at https://github.com/Henvezz95/nunchaku-fork.

3.4 Memory Optimization: Asymmetric KV-cache Quantization

Autoregressive generation is strictly memory-bound due to the monotonically growing Key-Value (KV) cache, which can consume over 40% of runtime memory in the Infinity 8B model. To resolve this bottleneck, we implement an **Asymmetric Per-channel INT8** strategy that halves the cache footprint.

Asymmetric Affine Quantization. We employ an affine mapping that preserves the skewness identified in Table 2:

$$\hat{x} = \text{clamp}\left(\left\lfloor \frac{x}{s} + z \right\rceil, -128, 127\right), \quad s = \frac{q_{hi} - q_{lo}}{255}, \quad z = \text{round}\left(-128 - \frac{q_{lo}}{s}\right), \tag{3}$$

where the scale $s \in \mathbb{R}^C$ and zero-point $z \in \mathbb{Z}^C$ are computed per-channel. We explicitly select asymmetric quantization ($z \neq 0$) because forcing a symmetric range on the highly skewed outlier channels ($\gamma > 7$) would cause significant information loss.

Optimal Clipping via Golden-section Search. Standard min-max quantization is sensitive to transient outliers. To maximize precision, we treat the clipping bounds $[q_{lo}, q_{hi}]$ as a hyperparameter optimization problem minimizing the reconstruction MSE. We parameterize the bounds via a percentile p and locate the optimum using a *coarse-to-fine strategy*: we first scan a logarithmic grid of percentiles ($p \in \{0.99, 0.999, \ldots, 1 - 10^{-6}\}$) to identify a candidate region, then refine the optimal p^* using golden-section search. This optimization strategy effectively balances the trade-off between clipping error and quantization resolution, allowing INT8 KV-cache quantization without any negative impact on generation quality.

4 Results and Analysis

4.1 Experimental Setup

Hardware & Proxy Methodology. All experiments were conducted on an NVIDIA Jetson AGX Orin 64GB Developer Kit. To rigorously assess deployment feasibility on resource-constrained edge devices, we utilize a proxy methodology: the high-capacity 64GB host allows us to benchmark uncompressed baselines that would otherwise trigger Out-Of-Memory (OOM) failures. We define three target memory envelopes corresponding to standard industrial modules: High-End (32GB, AGX Orin), Mid-Range (16GB, Orin NX), and Entry-Level (8GB, Orin Nano). A configuration is deemed feasible only if its total peak system footprint fits strictly within these physical limits.

Models & Datasets. We evaluate the Infinity family at two scales: Infinity-2B ($d_{model} = 2048$) and Infinity-8B ($d_{model} = 3584$). For validation, we utilize fixed 5 000-sample subsets from two distinct domains: MJHQ-30K [13] serves as a benchmark for high-fidelity artistic generation, while sDCI [24] (Densely Captioned Images) contains complex scene descriptions used to evaluate photo-realism and strict prompt adherence. Crucially, the calibration and evaluation

Table 3. Generative Quality vs. SOTA (MJHQ-30K & sDCI). We compare the Infinity models, compressed via our proposed pipeline (W4A4+KV8), against leading diffusion models in FP16 and SVDQuant W4A4 (INT4) formats. We measure ImageReward, CLIP Score, CLIP-IQA, and FID. Best quantized results are **bolded**.

Model	Config	MJHQ-30K (Artistic)				sDCI (Photorealism)			
		IR ↑	CLIP ↑	IQA ↑	FID ↓	IR ↑	CLIP ↑	IQA ↑	FID ↓
Diffusion Baselines									
SDXL	FP16	0.729	27.2	0.907	16.6	0.573	26.5	0.911	22.5
	W4A4	0.601	26.7	0.878	20.6	0.477	26.2	0.862	26.2
PixArt-Σ	FP16	0.944	26.8	0.944	16.6	0.966	26.1	0.966	24.8
	W4A4	0.878	26.6	0.926	19.2	0.918	26.1	0.948	25.9
SANA	FP16	0.952	26.8	0.934	20.6	0.847	26.4	0.958	29.9
	W4A4	0.935	26.9	0.926	19.3	0.846	26.4	**0.951**	28.1
Flux.1-dev	FP16	0.953	26.0	0.952	20.3	1.02	25.4	0.955	24.8
	W4A4	0.935	25.8	**0.950**	19.9	1.01	25.3	**0.951**	24.3
Visual Autoregressive									
Infinity 2B	FP16	0.981	25.5	0.947	21.3	0.997	25.3	0.955	28.6
	W4A4+KV8	0.840	25.5	0.919	20.2	0.828	25.2	0.927	25.5
Infinity 8B	FP16	1.18	27.1	0.945	19.6	1.17	26.9	0.947	21.6
	W4A4+KV8	**1.13**	**27.0**	0.935	**19.0**	**1.12**	**26.7**	0.925	**19.9**

sets are strictly disjoint; no prompts used for validation were seen during the quantization calibration phase. For calibration, we employ a step-aware strategy where we sample 12 diverse prompts for each of the 13 autoregressive generation steps. This results in a cumulative calibration set of $12 \times 13 = 156$ samples.

Baselines & Metrics. We establish a comparative framework comprising internal baselines (Block-Wise Quantization W4A4, SmoothQuant W4A4, SVDQuant W4A4 with rank 32 and W4A16 block-wise weight-only) and state-of-the-art external diffusion models, including Flux.1-dev [2], PixArt-Σ [3], Stable Diffusion XL (SDXL) [19] and SANA [32]. Evaluation covers three axes: (1) Generative Quality, using ImageReward [34] to quantify alignment with human aesthetic preferences, CLIP Score [10] for semantic text-image consistency, and CLIP-IQA [29]/FID [11] for perceptual fidelity; (2) Reference Fidelity, measuring pure quantization noise via PSNR, SSIM, and LPIPS [36] relative to the FP16 golden baseline; and (3) System Efficiency, tracking peak memory usage and end-to-end latency on the NVIDIA Jetson AGX Orin 64GB. All models are set to have an output resolution of $1\,024 \times 1\,024$.

Implementation Note. We employ W4A4 quantization globally, except for the cross-attention KV projection inputs, which are kept in FP16. As these inputs stem from the static text encoder, this hybrid precision strategy preserves prompt adherence with negligible system overhead, while their weights remain in INT4 to maximize storage efficiency.

4.2 Comparative Analysis: Generative Quality

We first establish that our quantization strategy preserves the state-of-the-art capabilities of the Infinity architecture against leading diffusion baselines. Ta-

ble 3 details the performance of Infinity VAR on the artistic (MJHQ-30K) and photorealistic (sDCI) benchmarks. The results reveal three critical insights regarding the deployability of VAR models:

1. Aesthetic Advantage. The most striking result is the performance of the quantized Infinity 8B model. Our W4A4+KV8 implementation achieves ImageReward scores of 1.13 and 1.12 across the MJHQ and sDCI datasets, respectively, significantly surpassing all evaluated baselines (both FP16 and quantized). Notably, it outperforms the widely adopted Flux.1-dev (12B parameters), which scores 0.953/1.02 in FP16 format. This indicates that even under aggressive compression, the 8B VAR architecture generates outputs that are consistently rated as more aesthetically pleasing than comparable diffusion models. The high quality of the generated images can be seen in Fig. 2.

The narrative for the lightweight Infinity 2B is more nuanced: in its FP16 state, it achieves a competitive ImageReward (0.981), rivaling uncompressed baselines like SANA. However, quantization exacts a heavier toll on this compact model (IR 0.981 $\rightarrow$ 0.840 on MJHQ). Despite this, both 2B and 8B models maintain exceptional CLIP-IQA scores (> 0.91), confirming that while the 2B model loses some stylistic preference, it retains high perceptual sharpness and technical fidelity.

2. Semantic Robustness & Scaling Laws. Contrary to the assumption that autoregressive models struggle with complex prompt following, the Infinity 8B demonstrates superior semantic consistency, achieving CLIP Scores of 27.0 and 26.7 on MJHQ and sDCI, respectively—the highest among all quantized models. This indicates that quantizing weights, activations, and the KV-cache does not degrade the ability to interpret nuanced instructions. Conversely, the Infinity 2B yields lower CLIP scores (25.5 on MJHQ, 25.2 on sDCI), but crucially, these scores remain stable compared to the FP16 baseline, confirming that semantic understanding is robust to the quantization process. Fig. 2 shows that overall prompt understanding is not affected by our quantization pipeline.

3. The Capacity Gap (2B vs. 8B). A distinct divergence is observed across model scales. While the Infinity 8B model retains nearly all of its aesthetic quality after quantization (IR 1.18 $\rightarrow$ 1.13 on MJHQ), the Infinity 2B model suffers a more noticeable drop (IR 0.981 $\rightarrow$ 0.840 on MJHQ). First, as detailed in Table 1, the 2B model exhibits significantly stronger activation outliers (353$\times$) than the 8B variant (226$\times$), making it inherently harder to compress. Second, the substantial parameter count of the 8B model likely provides the necessary redundancy to absorb the noise introduced by the quantization process.

Remark on FID: We observe that quantization yields a slight improvement in FID for both Infinity models. This phenomenon is not unique to our method and appears in other baselines (e.g., SANA). We attribute this to the fact that FID is computed against a reference dataset distinct from the training distribution. Therefore, we treat FID primarily as a sanity check against mode collapse rather than a strict quality ranking.

Fig. 2. Qualitative Comparison of W4A4 Quantization. We evaluate fidelity across four scenarios: Detailed Portrait, Architectural Geometry, Landscape Gradients, and Object Representation. Columns compare Infinity (2B/8B) in FP16 vs. our W4A4 quantization pipeline against SANA and Flux.1-dev (both quantized via SVDQuant W4A4 INT4). **Results:** The quantized Infinity 8B retains near-FP16 quality, often outperforming baselines in detail retention. Infinity 2B remains structurally comparable to SANA. While quantization introduces minor artifacts, our pipeline preserves global scene composition and high-frequency structure.

Prompts used: (1) "Portrait, photograph, Canon 5D, magazine, editorial, full profile shot, photorealism, Annie Leibovitz, middle-aged man, realistic, accurate"; (2) "A photograph of an intricate wooden gazebo with a traditional Asian-style tiled roof, set in a dense wooded forest clearing. Towering mountains in the background under a partly cloudy sky. Natural daylight, highly detailed wood grain and foliage, 8k"; (3) "Photorealistic. 4k. A hidden beach accessible only by boat, surrounded by towering rock formations, lush vegetation, and colorful coral reefs, the sun sets behind the mountains, a subtle warm orange glows over the water, creating a peaceful and romantic setting. Multiple light sources, high detail, ultra-realistic"; (4) "A detailed close-up photograph of a small shrine against a solid black background. A weathered stone statue in the center, surrounded by fresh colorful flowers, several burning candles casting warm light, and fruit wrapped in clear cellophane plastic. Macro lens, highly textured, cinematic lighting, 8k". (1) and (4) are taken from MJHQ, (2) and (3) from sDCI. The same seed is used for all models.

4.3 System Efficiency: Breaking the Memory Wall

Measurement Protocol. Table 4 reports the Peak System Memory, defined as the maximum allocated footprint including model weights, KV-cache, VAE, text-encoder, and activation buffers during generation. Latency measures the end-to-end generation time for a $1\,024 \times 1\,024$ image (batch size 1). The Feasible HW column identifies the minimum commercial NVIDIA Jetson module capable of hosting the model without swapping.

Table 4. System Efficiency vs. SOTA (Edge Constraints). We compare the resource requirements of our quantized Infinity models against leading diffusion models (FP16 and SVDQuant versions) and show the minimum hardware requirements for each of them. Latency was measured on an NVIDIA Jetson AGX Orin 64GB.

Model	Method	Peak Memory	Latency	Feasible HW
Flux.1-dev	FP16	33.8 GB	246 s	AGX Orin (64GB)
	W4A4	11.8 GB	112 s	Orin NX (16GB)
SANA	FP16	10.2 GB	9.84 s	Orin NX (16GB)
	W4A4	7.19 GB	6.05 s	*Orin Nano (8GB)*
Infinity 2B	FP16	16.0 GB	8.46 s	AGX Orin (32GB)
	W4A4	12.3 GB	9.52 s	Orin NX (16GB)
	W4A4+KV8	**7.71 GB**	**11.5 s**	**Orin Nano (8GB)**
Infinity 8B	FP16	37.1 GB	25.1 s	AGX Orin (64GB)
	W4A4	23.6 GB	22.9 s	AGX Orin (32GB)
	W4A4+KV8	**13.3 GB**	**27.0 s**	**Orin NX (16GB)**

Breaking the Hardware Barrier: From Server to Edge. The most critical impact of our proposed quantization pipeline is the fundamental shift it enables in deployment economics. Standard FP16 inference for the Infinity 8B demands 37.1GB of system memory, strictly confining it to data-center-class GPUs or the top-tier AGX Orin 64GB (∼$2 000). As shown in Table 4, our pipeline shatters this barrier, reducing the footprint by 64% to just 13.3GB. This creates a new feasibility tier for industrial deployment: the flagship 8B model can now run locally on the mid-range Orin NX 16GB (∼$600), effectively reducing the hardware cost by 70%. Similarly, we compress the Infinity 2B from 16.0GB down to 7.71GB, unlocking deployment on the entry-level Orin Nano (8GB).

Comparative Efficiency & The Latency Advantage. Our method establishes a new efficiency standard against diffusion baselines. While Flux.1-dev (SVDQuant W4A4) achieves a comparable memory footprint (11.8GB), its heavy transformer architecture remains computationally demanding, requiring 112 seconds on the Jetson AGX Orin to generate an image. Our quantized Infinity 8B generates samples in 27.0 seconds (4.1× speedup), making it the only high-fidelity, memory-feasible, and temporally viable model for edge applications.

Implementation Analysis: Python vs. Kernels. We note a divergence in latency trends: diffusion baselines (Flux, SANA) speed up with quantization (e.g., SANA $9.8s \rightarrow 6.0s$) due to the use of full pre-compiled C++/CUDA architectures available on Nunchaku. Conversely, our Infinity implementation sees a slight latency increase ($25.1s \rightarrow 27.0s$ for the 8B model). We attribute this overhead to two sources: (1) the Python-level wrapper for weight quantization, which introduces latency and masks the raw INT4 arithmetic gains; and (2) the online quantization of the KV-cache, which adds ∼4.1 seconds to the generation loop on the 8B model (comparing W4A4 vs. W4A4+KV8). This suggests that while our current implementation is sufficient to demonstrate deployment feasibility, rewriting the entire model architecture in custom C++/CUDA kernels would further reduce latency, since replacing high-level sequential logic with direct management of memory accesses and thread-level execution is often strictly required to fit GPU logic and unlock true hardware acceleration [1].

Table 5. Ablation Study. We evaluate the impact of different quantization strategies on the Infinity architecture. The baseline is block-wise quantization with $B = 64$. The standard W4A4 baseline causes significant degradation in the 2B model (IR 0.981 $\rightarrow$ 0.616). The application of SVDQuant recovers fidelity (IR 0.848). SVDQuant + KV8 integrates asymmetric cache compression, without reducing output quality further.

Configuration	Generative Metrics				Reference Metrics		
	IR ↑	CLIP ↑	CLIP-IQA ↑	FID ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Infinity 2B - MJHQ Dataset							
FP16 (Golden)	0.981	25.53	0.947	21.30	-	-	-
Baseline W4A4	0.616	25.11	0.881	20.70	11.04	0.424	0.669
Baseline W4A16	0.892	25.30	0.936	20.92	12.52	0.485	0.566
SmoothQuant W4A4	0.667	25.34	0.899	19.80	11.37	0.449	0.656
SVDQuant W4A4	0.848	25.45	0.919	20.28	12.12	0.464	0.625
Ours (SVD + KV8)	0.840	25.46	0.919	20.21	12.13	0.467	0.625
Infinity 8B - MJHQ Dataset							
FP16 (Golden)	1.180	27.06	0.946	19.63	-	-	-
Baseline W4A4	1.120	26.89	0.932	18.73	13.09	0.453	0.496
SmoothQuant W4A4	1.107	26.92	0.930	18.97	13.08	0.451	0.500
SVDQuant W4A4	1.129	26.96	0.933	18.99	13.22	0.457	0.487
Ours (SVD + KV8)	1.128	26.96	0.935	18.99	13.20	0.454	0.487

The Entry-level Tier. Finally, for the strict constraints of the Orin Nano (8GB), our Infinity 2B provides a robust autoregressive alternative to SANA (7.19GB). We match SANA's memory efficiency while retaining the distinct architectural advantages of the VAR paradigm, demonstrating that high-resolution autoregressive generation is viable even on entry-level edge silicon. While real-time industrial perception has numerous demonstrations in the academic literature (e.g., [26, 28]), our pipeline pushes edge hardware to its limits by enabling billion-scale generative modeling.

4.4 Ablation Studies

The Activation Bottleneck. The compact 2B model serves as a rigorous stress test for quantization methods. As shown in Table 5, the block-wise W4A4 baseline suffers a catastrophic collapse (ImageReward 0.981 $\rightarrow$ 0.616), producing incoherent outputs. Crucially, the W4A16 (Weight-Only) control retains high fidelity (IR 0.892), effectively decoupling the error sources: since 4-bit weights perform well, the failure of W4A4 is mostly attributable to activation outliers. While SmoothQuant provides a marginal improvement over the block-wise quantization baseline (IR 0.667), it fails to fully mitigate the massive outliers intrinsic to the architecture. In contrast, SVDQuant successfully migrates these outliers to the weights, recovering an ImageReward of 0.848 and approaching the upper bound established by the W4A16 control. The 8B model exhibits greater inherent robustness; even the baseline W4A4 yields usable outputs (IR 1.120). However, the method hierarchy remains consistent: *Block-Wise* < *SmoothQuant* < *SVDQuant*. The SVD-based approach (IR 1.129) still outperforms baselines.

The Cost of KV-cache Compression. Finally, we isolate the impact of our Asymmetric KV8 strategy. Comparing the SVDQuant rows (W4A4) against

Ours (W4A4+KV8) reveals no meaningful degradation in either aesthetic quality or reference fidelity across both model scales. For Infinity 8B, the drop is virtually non-existent (IR 1.129 $\rightarrow$ 1.128), and for Infinity 2B, metrics remain stable (0.848 $\rightarrow$ 0.840), confirming KV-cache redundancy in VAR models and allowing for aggressive compression without influencing the generative outcome.

5 Conclusion

In this work, we addressed the "Memory Wall" constraining visual autoregressive models on edge devices. We systematically diagnosed the Infinity VAR architectures (2B and 8B models), identifying extreme outliers in the activations reaching 353$\times$ the median. Our analysis also showed that the KV-cache exhibits a significantly higher coefficient of variation across channels compared to tokens, with specific channels exhibiting strong skewness. From these insights, we developed a specialized compression pipeline, combining INT4 W4A4 SVDQuant for weights and activations with asymmetric per-channel INT8 quantization for the cache. This approach enables the flagship Infinity-8B model to run on mid-range hardware (Orin NX 16GB) by reducing the memory footprint by 64% (37.1GB $\rightarrow$ 13.3GB), effectively lowering deployment hardware costs by approximately 70%. Crucially, this efficiency is achieved without compromising fidelity; our quantized 8B model attains an ImageReward of 1.13 on MJHQ, surpassing the 12B Flux.1-dev baseline (IR 0.935) while generating images 4.1$\times$ faster, demonstrating that high-fidelity autoregressive generation is feasible on the edge, offering a superior quality-latency trade-off compared to state-of-the-art diffusion transformers.

References

1. Allegretti, S., et al.: How does Connected Components Labeling with Decision Trees perform on GPUs? In: CAIP (2019)
2. Black Forest Labs: Flux.1. https://blackforestlabs.ai/ (2024)
3. Chen, J., et al.: PixArt-Σ: Weak-to-Strong Training of Diffusion Transformer for 4K Text-to-Image Generation. In: ECCV (2024)
4. Dettmers, T., et al.: GPT3.int8(): 8-bit Matrix Multiplication for Transformers at Scale. NeurIPS (2022)
5. Esser, P., Rombach, R., Ommer, B.: Taming Transformers for High-Resolution Image Synthesis. In: CVPR (2021)
6. Firoozi, R., et al.: Foundation models in robotics: Applications, challenges, and the future. The International Journal of Robotics Research **44**(5) (2025)
7. Frantar, E., et al.: GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. arXiv:2210.17323 (2022)
8. Golda, A., et al.: Privacy and Security Concerns in Generative AI: A Comprehensive Survey. IEEE Access **12** (2024)
9. Han, J., et al.: Infinity: Scaling Bitwise AutoRegressive Modeling for High-Resolution Image Synthesis. In: CVPR (2025)
10. Hessel, J., et al.: CLIPScore: A Reference-free Evaluation Metric for Image Captioning. In: EMNLP (2021)

11. Heusel, M., et al.: GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. NeurIPS (2017)
12. Lee, D., et al.: Autoregressive Image Generation using Residual Quantization. In: CVPR (2022)
13. Li, D., et al.: Playground v2.5: Three Insights towards Enhancing Aesthetic Quality in Text-to-Image Generation. arXiv:2402.17245 (2024)
14. Li, K., et al.: Memory-Efficient Visual Autoregressive Modeling with Scale-Aware KV Cache Compression. arXiv:2505.19602 (2025)
15. Li, M., et al.: SVDQuant: Absorbing Outliers by Low-Rank Components for 4-Bit Diffusion Models. arXiv:2411.05007 (2024)
16. Lin, J., et al.: AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration. MLSys **6** (2024)
17. Liu, Z., et al.: KIVI: A Tuning-Free Asymmetric 2bit Quantization for KV Cache. arXiv:2402.02750 (2024)
18. Mentzer, F., et al.: High-Fidelity Generative Image Compression. NeurIPS (2020)
19. Podell, D., et al.: SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. arXiv preprint arXiv:2307.01952 (2023)
20. Qin, Z., et al.: Head-Aware KV Cache Compression for Efficient Visual Autoregressive Modeling. arXiv:2504.09261 (2025)
21. Razavi, A., Van den Oord, A., Vinyals, O.: Generating Diverse High-Fidelity Images with VQ-VAE-2. NeurIPS (2019)
22. Tang, H., et al.: HART: Efficient Visual Generation with Hybrid Autoregressive Transformer. arXiv:2410.10812 (2024)
23. Tian, K., et al.: Visual Autoregressive Modeling: Scalable Image Generation via Next-Scale Prediction. NeurIPS (2024)
24. Urbanek, J., et al.: A Picture is Worth More Than 77 Text Tokens: Evaluating CLIP-Style Models on Dense Captions. In: CVPR (2024)
25. Van Den Oord, A., et al.: Neural Discrete Representation Learning. NeurIPS (2017)
26. Vezzali, E., et al.: A Deep-Learning-Based Method for Real-Time Barcode Segmentation on Edge CPUs. In: CAIP (2025)
27. Vezzali, E., et al.: Mosaic-SR: An Adaptive Multi-step Super-Resolution Method for Low-Resolution 2D Barcodes. In: ICIP (2025)
28. Vezzali, E., et al.: State-of-the-art review and benchmarking of barcode localization methods. Engineering Applications of Artificial Intelligence (2025)
29. Wang, J., Chan, K.C., Loy, C.C.: Exploring CLIP for Assessing the Look and Feel of Images. In: AAAI. vol. 37 (2023)
30. Wei, R., et al.: FPQVAR: Floating Point Quantization for Visual Autoregressive Model with FPGA Hardware Co-design. arXiv:2505.16335 (2025)
31. Xiao, G., et al.: SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models. In: ICML (2023)
32. Xie, E., et al.: SANA: Efficient High-Resolution Image Synthesis with Linear Diffusion Transformers. arXiv preprint arXiv:2410.10629 (2024)
33. Xie, R., et al.: LiteVAR: Compressing Visual Autoregressive Modelling with Efficient Attention and Quantization. arXiv:2411.17178 (2024)
34. Xu, J., et al.: ImageReward: Learning and Evaluating Human Preferences for Text-to-Image Generation. NeurIPS (2023)
35. Yu, J., et al.: Vector-quantized Image Modeling with Improved VQGAN. arXiv:2110.04627 (2021)
36. Zhang, R., et al.: The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In: CVPR (2018)