QUANTIZED KEYS STEAL ATTENTION: BIAS CORRECTION FOR KV-CACHE COMPRESSION IN VIDEO DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Chunk-wise autoregressive video diffusion models rely on a KV cache of previously generated chunks to avoid redundant computation, and this cache quickly becomes a memory bottleneck as videos grow longer. Methods that quantize the KV cache to low bitwidths reduce memory pressure but degrade video quality. We show that a key driver of this degradation is a systematic bias in attention weights: due to the convexity of the exponential in softmax attention, quantization noise inflates the contribution of cached keys, a phenomenon we call the *Jensen bias*. This effect causes quantized keys to steal attention mass from the unquantized current chunk. We derive a per-attention-score correction that removes this bias in expectation, computed on the fly from the quantization step sizes of the cached keys and the query norm. With a second-order Taylor approximation, the correction adds negligible computational overhead and needs no additional memory alongside the cache. Evaluated on MAGI-1, SkyReels-V2, and HY-WorldPlay at INT2 quantization, our correction recovers most of the quality lost to aggressive quantization, reaching near-BF16 video quality, and can match or exceed INT4 quantization while using 50% less memory. Because it removes the residual Jensen bias rather than the quantization error itself, the correction complements upstream quantizers such as QuantVideoGen.

1 INTRODUCTION

Video diffusion models have made remarkable progress in generating short, high-fidelity clips (Yang et al., 2025; Kong et al., 2025; Team Wan et al., 2025). Recent work on video generation models has introduced chunk-wise autoregressive video diffusion (Chen et al., 2024), where each chunk of frames is denoised in turn and attends to previously generated chunks (Yin et al., 2025; Sand.ai et al., 2025; Chen et al., 2025; Sun et al., 2025). To avoid recomputing the key and value representations of past chunks at every denoising step, autoregressive models store them in a KV cache and reuse them across subsequent chunks. In this setting, the KV cache acts as the model's temporal memory: it determines how much previously generated visual context remains available when simulating the next chunk of a video or world trajectory.

To further reduce the attention cost, MAGI-1 (Sand.ai et al., 2025) attends to a sliding window of the last n cached chunks, yielding linear instead of quadratic scaling in video length. This design introduces a fundamental memory–context trade-off: increasing the window size improves temporal consistency by providing more past context, but also increases the size of the KV cache proportionally. Due to memory capacity, memory bandwidth, and latency constraints in practical systems, the window size must be limited, restricting the temporal information available to the model and degrading long-range consistency (Xi et al., 2026; Samuel et al., 2026).

KV-cache quantization directly targets the underlying memory bottleneck by compressing the cached keys and values to lower bitwidths, thereby relaxing this trade-off: the same memory budget can support a larger context window, or a fixed window can be stored more efficiently. Prior work on KV-cache quantization for LLM inference (Liu et al., 2024; Hooper et al., 2024; Ashkboos et al., 2024) has established effective techniques down to 2-bit precision. For autoregressive video models,

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.



Figure 1: Qualitative comparison on MAGI-1 for two representative prompts. Columns show successive frames from the same generated video. From top to bottom: BF16 baseline; asymmetric INT2 (QuaRot+RTN) KV-cache quantization of both keys and values; same quantized setting with our correction. INT2 quantization quickly destroys subject and scene structure, whereas our correction substantially recovers the BF16-like visual quality and temporal consistency.

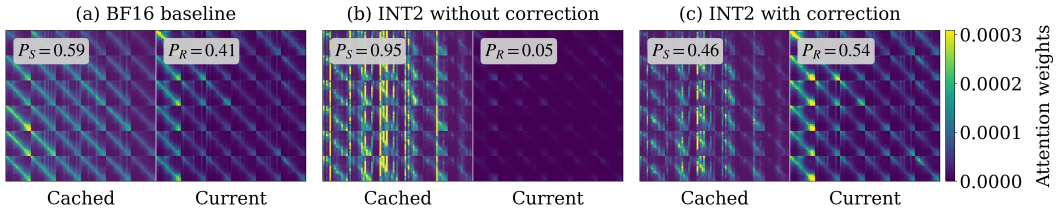


Figure 2: Attention weights for MAGI-1 for the prompt "a person" under INT2 KV-cache quantization. The visualization is taken from a representative layer, time step, and attention head. Panel **(b)** shows that relative to the BF16 baseline in **(a)**, quantization increases attention weights in the cached block of tokens and decreases them in the current chunk. This effect is quantified by the *attention masses* $P_{\mathcal{S}}$ and $P_{\mathcal{R}} = 1 - P_{\mathcal{S}}$ of the cached token block and the current chunk. **(c)** shows that our correction largely restores the original attention weights.

we find that INT4 KV-cache quantization preserves reasonable quality, whereas reducing to INT2 with round-to-nearest quantization leads to severely distorted frames (Fig. 1, Fig. 8, and Fig. 9).

We identify a shift of *attention mass* toward cached tokens under aggressive quantization as an important source of this degradation (see example in Fig. 2 and definition in Section 4.1). This shift is consistent across layers, heads, denoising steps, and prompts, and correlates with attention-output error on the same heads (Appendix E.4); removing it recovers video quality (Section 5). Integer quantization introduces approximately zero-mean noise into the cached keys, leaving pre-softmax attention scores unbiased in expectation. However, the exponential in softmax breaks this symmetry: due to its convexity, positive deviations are amplified more than equally large negative deviations are suppressed. As a result, a symmetric score-level noise distribution becomes right-skewed after exponentiation, with its mean systematically exceeding the exponential of the original unquantized score (Fig. 3). We refer to this systematic, convexity-induced inflation as the *Jensen bias*, as it is an instance of the Jensen gap studied in probability theory (Gao et al., 2020). In chunk-wise autoregressive video diffusion, this bias inflates the cached-token contribution to the softmax partition sum at the expense of the current chunk.

Our correction directly targets the Jensen bias. Because the bias is systematic, it can be estimated from quantities available at inference time and subtracted from the cached-key attention scores before the softmax. This restores the balance between cached and current tokens without retraining or modifying the quantized KV-cache values (Fig. 2).

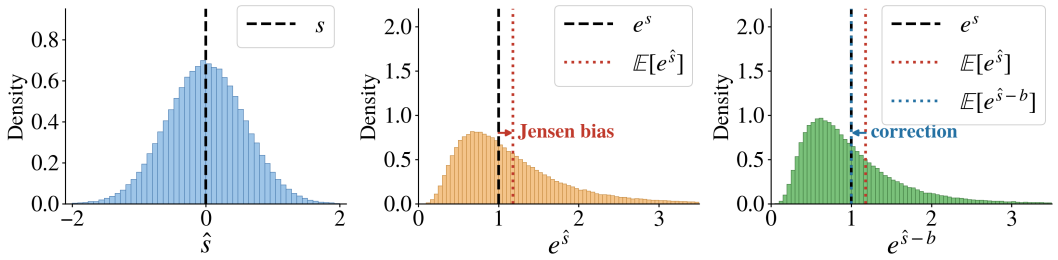


Figure 3: Illustration of the *Jensen bias* and its correction on a single attention score. **Left:** Quantization noise $\delta \sim \mathcal{N}(0, \sigma^2)$ with zero mean produces a noisy score $\hat{s} = s + \delta$ centered at s. **Center:** After exponentiation the distribution becomes right-skewed: its mean $\mathbb{E}[e^{\hat{s}}]$ strictly exceeds e^s by the so-called *Jensen bias*. **Right:** Subtracting a correction b shifts the mean $\mathbb{E}[e^{\hat{s}-b}]$ closer to e^s, largely removing the systematic Jensen bias.

Our contributions are as follows:

- We identify the Jensen bias: zero-mean quantization noise in cached keys systematically inflates their expected softmax contribution and thus shifts attention mass away from the unquantized current chunk. To our knowledge, this effect has not been reported before.
- We derive a per-attention-score correction that is exact for any noise distribution, and a second-order Taylor approximation with negligible overhead that needs no retraining, calibration data, extra metadata, or cache modification.
- We demonstrate consistent benchmark improvements across models and quantization schemes, from attention-level diagnostics to end-to-end video quality, and show that the correction composes with upstream quantizers such as QuantVideoGen.

2 Related Work

KV-cache quantization for LLMs. The KV cache is a well-known memory bottleneck in long-context LLM inference (Kwon et al., 2023), and a growing body of work addresses it through quantization: KIVI (Liu et al., 2024) provides an early systematic study of KV-cache element distributions, observing that keys exhibit channel-wise outliers while values do not, and exploits this asymmetry to achieve tuning-free 2-bit KV quantization. KVQuant (Hooper et al., 2024) combines per-channel key quantization with non-uniform datatypes calibrated to the empirical KV distribution and explicit isolation of outlier entries, pushing KV caches below 4 bits with minimal perplexity loss. QuaRot (Ashkboos et al., 2024) applies Hadamard rotations to spread channel-wise outliers before quantization, enabling outlier-free 4-bit inference. TurboQuant (Zandieh et al., 2025) similarly leverages random rotations, framing KV-cache compression as an online vector quantization problem and applying scalar quantization in the rotated space to achieve near-optimal distortion at low bitwidth. MixKVQ (Zhang et al., 2025b) uses query-aware mixed precision to retain important key channels at higher precision, selecting those channels by both their intrinsic quantization difficulty and their relevance to the query. AsymKV (Tao et al., 2024) observes that model loss is more sensitive to key quantization than value quantization and proposes layer-wise asymmetric bit allocation, supporting our choice to correct only the bias induced by key quantization (Section 5.3). Our work is orthogonal to the approaches above in that we do not improve the quantization scheme itself, but instead analytically correct the systematic bias in the attention weights introduced by any such scheme.

Attention sensitivity and correction. Several works have studied how quantization and other perturbations affect the attention mechanism. Pandey et al. (2023) show that quantizing the softmax computation introduces a large bias in the softmax output, degrading accuracy in generative models, and propose an offline correction that can be folded into the quantization parameters. Our work targets a different source of bias, focusing on KV-cache quantization rather than softmax quantization. KVLinC (Saxena & Roy, 2025) is conceptually closest to our approach: it introduces trainable linear correction adapters to compensate for errors from quantized keys. In contrast, our correction is training-free and analytically derived. SageAttention2 (Zhang et al., 2025a) smooths queries

by subtracting channel means and adds a correction term to the scores. However, this targets quantization-friendliness of the $QK^\top$ product rather than the systematic bias from exponentiation. Yao et al. (2024) propose timestep-aware corrections for quantized diffusion models, demonstrating that structure-aware corrections can substantially reduce quantization degradation, a principle our per-attention-score correction shares.

Autoregressive video diffusion and efficient caching. Building on diffusion forcing (Chen et al., 2024), chunk-wise autoregressive video diffusion models generate videos by denoising successive chunks that attend to previously generated chunks through a KV cache (Yin et al., 2025; Sand.ai et al., 2025; Chen et al., 2025; Sun et al., 2025). Because the cache grows with each new chunk, a growing body of work aims to reduce its cost through cache compression and eviction (Ma et al., 2026; Chen et al., 2026a; Samuel et al., 2026), sparse attention (Lv et al., 2026), or direct quantization of the cached states (Xi et al., 2026). Among these, QuantVideoGen (Xi et al., 2026) is most directly related to our approach: it applies training-free KV-cache quantization using semantic-aware smoothing and progressive residual quantization to reduce the quantization error itself. Our approach is complementary: rather than reducing the quantization error, we analytically correct the bias it introduces in softmax attention, so the two methods can be combined (Table 1).

3 PRELIMINARIES

Integer quantization. Integer quantization maps a floating-point value to a discrete grid defined by a *scale* Δ, also known as the step size between adjacent grid levels, and a *zero-point* z. Given a B-bit quantization target, each element x is mapped to $x_q = \mathrm{clamp}(\lfloor x/\Delta \rceil + z,\, 0,\, 2^B{-}1)$, where $\lfloor\cdot\rceil$ denotes rounding to nearest (RTN), and is reconstructed as $\hat{x} = (x_q - z)\cdot\Delta$. The round-trip $x \mapsto x_q \mapsto \hat{x}$ introduces an additive error $\epsilon = \hat{x} - x$. Since Δ and z are chosen to cover the full $[\min, \max]$ range of the values being quantized, almost no value is clipped (Appendix B), and the error is bounded by $|\epsilon| \le \Delta/2$. Our default is group-wise per-token quantization, which divides the d channels of each token i into $G = d/g$ groups of size g, with an independent $(\Delta_{i,j}, z_{i,j})$ per group j. Smaller groups reduce the quantization error but increase the memory footprint.

Hadamard rotation. Key vectors in transformer models often exhibit channel-wise outliers, i.e. a few channels have much larger magnitudes than the rest (Dettmers et al., 2022; Ashkboos et al., 2024). These outliers inflate the quantization step size Δ, degrading precision for all other channels. QuaRot (Ashkboos et al., 2024) spreads the outlier energy across all channels by applying a randomized Hadamard rotation $H \in \mathbb{R}^{d\times d}$ (with $H^\top H = I$) to both keys and queries. The resulting distribution is more uniform, allowing for lower quantization errors. Because H is orthogonal, the attention scores are preserved: $(Hq)^\top(Hk) = q^\top k$.

Token structure and attention decomposition. In autoregressive video diffusion, each chunk of video frames is encoded into a latent representation and patchified into a grid of spatio-temporal tokens before entering the transformer. Depending on the model and resolution, this results in several thousand tokens per chunk. At each denoising step, every query in the current chunk attends to two groups of keys: (i) the keys of the current chunk, which are computed in full precision at every step, and (ii) the keys of previously generated chunks, which were written to a KV cache once each chunk finished denoising and are reused without recomputation. The attention score matrix therefore decomposes into two blocks: a *current* block of tokens (current-chunk queries $\times$ current-chunk keys) and a *cached* block of tokens (current-chunk queries $\times$ cached keys).

4 METHOD

We show that KV-cache quantization systematically inflates the softmax contribution of cached keys and derive a correction that removes this bias in expectation, together with a practical approximation.

4.1 Quantization Bias in Softmax Attention

Consider a single attention head with dimension d. For a query vector $q \in \mathbb{R}^d$ and key vectors $k_i \in \mathbb{R}^d$, where i is the token index, the attention score and attention weight for token i are

$$s_i = \frac{q^\top k_i}{\sqrt{d}}, \qquad p_i = \frac{e^{s_i}}{\sum_{j=1}^{N} e^{s_j}}. \tag{1}$$

Following the token structure in Section 3, let $\mathcal{S}$ denote the set of quantized *cached* key indices and $\mathcal{R}$ the set of unquantized *current-chunk* key indices, so that $\{1, \ldots, N\} = \mathcal{S} \cup \mathcal{R}$. We define the partition sums and the total attention mass on the cached block,

$$Z_{\mathcal{S}} = \sum_{i \in \mathcal{S}} e^{s_i}, \qquad Z_{\mathcal{R}} = \sum_{i \in \mathcal{R}} e^{s_i}, \qquad P_{\mathcal{S}} = \sum_{i \in \mathcal{S}} p_i = \frac{Z_{\mathcal{S}}}{Z_{\mathcal{S}} + Z_{\mathcal{R}}}, \tag{2}$$

which measures how much attention mass is assigned to cached keys, and is what we ultimately care about when analyzing attention stealing. For a representative example of attention stealing, compare panels (a) and (b) of Fig. 2.

Quantization noise in attention scores. Let $\Delta_{i,c}$ denote the quantization step size for channel c of cached token i. The quantize–dequantize round-trip yields $\hat{k}_i = k_i + \epsilon_i$ for $i \in \mathcal{S}$, where $\epsilon_i \in \mathbb{R}^d$ is the per-element error of integer quantization. The quantized attention score is then

$$\hat{s}_i = \frac{q^\top \hat{k}_i}{\sqrt{d}} = s_i + \delta_i, \qquad \delta_i = \frac{q^\top \epsilon_i}{\sqrt{d}}, \tag{3}$$

where δ_i is the attention-score noise for key i. For unquantized keys $i \in \mathcal{R}$, we have $\hat{s}_i = s_i$.

Jensen bias and attention stealing. Consider the quantized cached partition sum $\hat{Z}_{\mathcal{S}} = \sum_{i \in \mathcal{S}} e^{s_i + \delta_i}$. By linearity of expectation:

$$\mathbb{E}\big[\hat{Z}_{\mathcal{S}}\big] = \sum_{i \in \mathcal{S}} e^{s_i} \cdot \mathbb{E}\big[e^{\delta_i}\big]. \tag{4}$$

For each term, Jensen's inequality applied to the convex function $\exp(\cdot)$ gives $\mathbb{E}[e^{\delta_i}] \geq e^{\mathbb{E}[\delta_i]}$. Round-to-nearest integer quantization error is zero-mean under the standard additive noise model of Widrow et al. (1996), an assumption we adopt in Section 4.2 and verify empirically in Appendix B. Hence $\mathbb{E}[\delta_i] = 0$, the right-hand side of the inequality equals 1, and $\mathbb{E}[\hat{Z}_{\mathcal{S}}] \geq Z_{\mathcal{S}}$. We call this systematic inflation of $\hat{Z}_{\mathcal{S}}$ caused by δ_i the *Jensen bias*. See Fig. 3 for an illustration of this bias and its correction on a single attention score value. Since $Z_{\mathcal{R}}$ is unaffected by key quantization, inflation of $\hat{Z}_{\mathcal{S}}$ can shift attention mass toward cached keys. We quantify this *attention stealing* as

$$\Delta P_{\mathcal{S}} = \hat{P}_{\mathcal{S}} - P_{\mathcal{S}}, \qquad \hat{P}_{\mathcal{S}} = \frac{\hat{Z}_{\mathcal{S}}}{\hat{Z}_{\mathcal{S}} + Z_{\mathcal{R}}}. \tag{5}$$

Positive values indicate excess attention on the cached block, as observed in Section 5.3.

4.2 Correction of the Jensen Bias

We derive a per-attention-score correction b_i that counteracts the Jensen bias, applied only to cached scores ($i \in \mathcal{S}$) and leaving current-chunk scores s_i ($i \in \mathcal{R}$) unchanged. As shown in Section 4.1, each cached token's contribution to the partition sum is individually biased upward: $\mathbb{E}[e^{s_i + \delta_i}] = e^{s_i}\,\mathbb{E}[e^{\delta_i}] \geq e^{s_i}$. We correct each token individually by requiring its expected contribution to match the unquantized value:

$$e^{s_i - b_i} \cdot \mathbb{E}\big[e^{\delta_i}\big] \stackrel{!}{=} e^{s_i} \quad \Longrightarrow \quad \boxed{b_i = \log \mathbb{E}\big[e^{\delta_i}\big].} \tag{6}$$

Let $\tilde{s}_i = \hat{s}_i - b_i$ denote the corrected score and $\tilde{Z}_{\mathcal{S}} = \sum_{i \in \mathcal{S}} e^{\tilde{s}_i}$ the corrected cached partition sum (a hat marks quantized quantities, a tilde quantized and corrected ones). Since every term is individually unbiased, the corrected cached partition sum is unbiased by linearity of expectation:

$$\mathbb{E}\big[\tilde{Z}_{\mathcal{S}}\big] = \sum_{i \in \mathcal{S}} e^{s_i - b_i} \cdot \mathbb{E}\big[e^{\delta_i}\big] = \sum_{i \in \mathcal{S}} e^{s_i} = Z_{\mathcal{S}}. \tag{7}$$

At inference time, we apply this correction by subtracting b_i from each cached attention score $\hat{s}_i$ prior to the softmax, leaving scores from the current (unquantized) keys unchanged. Note that $b_i \geq 0$ always (since $\mathbb{E}[e^{\delta_i}] \geq 1$ by Jensen's inequality).

The correction in Eq. (6) is exact for any noise distribution. Because evaluating the expectation requires the distribution of δ_i, we obtain a practical form in three steps, each of which we validate directly in Appendix B. For any random variable X with cumulants $\kappa_1, \kappa_2, \kappa_3, \ldots$, the cumulant generating function (CGF) has the Taylor expansion

$$K_X(t) = \log \mathbb{E}[e^{tX}] = \kappa_1 t + \frac{\kappa_2}{2} t^2 + \frac{\kappa_3}{6} t^3 + \cdots, \tag{8}$$

and $b_i = K_{\delta_i}(1)$. (i) We keep only the first two terms of this expansion at $t = 1$. This second-order Taylor approximation of the CGF gives $b_i \approx \mathbb{E}[\delta_i] + \frac{1}{2}\operatorname{Var}(\delta_i)$, which is accurate when the higher cumulants of the score noise are small. (ii) We adopt the standard additive noise model of round-to-nearest quantization (Widrow et al., 1996), under which the per-element key error $\epsilon_{i,c}$ is zero-mean. Hence $\mathbb{E}[\delta_i] = 0$ and $b_i \approx \sigma_i^2/2$ with $\sigma_i^2 = \operatorname{Var}(\delta_i)$. The correction is thus half the score-space noise variance, and it depends on the noise distribution only through its variance. (iii) We compute this variance. Under the same model, the components of ϵ_i are independent across channels $c \in \{1, \ldots, d\}$ and uniformly distributed, $\epsilon_{i,c} \sim \mathcal{U}(-\Delta_{i,c}/2,\ \Delta_{i,c}/2)$. Each $\epsilon_{i,c}$ then has variance $\Delta_{i,c}^2/12$, and since δ_i is a sum of independent terms, the variances add:

$$\sigma_i^2 = \operatorname{Var}(\delta_i) = \frac{1}{12\,d} \sum_{c=1}^{d} q_c^2\, \Delta_{i,c}^2. \tag{9}$$

Together, these steps yield the correction below, where the last equality holds for group-wise step sizes $\Delta_{i,j}$ (Section 3) with per-group squared query norm $\|q_j\|^2 = \sum_{c \in \text{group } j} q_c^2$:

$$\boxed{b_i \approx \frac{\sigma_i^2}{2} = \frac{1}{24\,d} \sum_{c=1}^{d} q_c^2\, \Delta_{i,c}^2 = \frac{1}{24\,d} \sum_{j=1}^{G} \Delta_{i,j}^2\, \|q_j\|^2}. \tag{10}$$

This second-order Taylor correction is simple and interpretable. It shows that the bias scales with both the squared query magnitude and the squared quantization step size, both of which are available at inference time. For a representative example of this proposed correction, compare panels (b) and (c) of Fig. 2. We use this approximation in all experiments. The same correction extends to QuaRot by replacing q with the rotated query Hq, so that $\|q_j\|^2$ becomes $\|(Hq)_j\|^2$ in Eq. (10), as shown in Appendix A.1.

Extension to other quantizers and formats. Channel independence and uniform key error are used only in the variance computation (step (iii)), so the method can be carried over to other zero-mean quantizers by supplying a different variance model or a measured $\operatorname{Var}(\delta_i)$. For floating-point formats such as FP, MXFP (Rouhani et al., 2023), and NVFP (NVIDIA et al., 2026), whose rounding error is proportional to the magnitude of the quantized value but can be described by approximate additive noise models (Widrow et al., 1996), one substitutes the corresponding score-space variance into $b_i \approx \sigma_i^2/2$ (Appendix A.2).

4.3 Effective Bitwidth and Runtime Overhead

For group-wise quantization with group size g, the effective bitwidth is $B_{\text{eff}} = B + \frac{24}{g}$, accounting for a per-group scale stored in FP8 and a zero-point stored in BF16. Our correction reuses these per-group scales and introduces no additional storage. In terms of computation, it adds an $O(QK \cdot d/g)$ term to attention, compared to the standard $O(QK \cdot d)$ cost of $QK^\top$, so the additional work is smaller by a factor of g. With our Triton kernel implementation, the correction adds 7.2% latency to the attention kernel on average, which corresponds to an estimated 1.8% end-to-end slowdown for MAGI-1. Peak GPU memory is unchanged, so the full INT2 memory saving is preserved. For more details about these storage and computation costs, see Appendices C.1 and C.4.

Table 1: Effect of the proposed correction for MAGI-1, SkyReels-V2, and HY-WorldPlay. The correction consistently improves fidelity (PSNR, SSIM, LPIPS) and perceptual quality (VBench), recovering much of the degradation introduced by quantization. Fidelity metrics are computed against the BF16 outputs. RTN and QuaRot+RTN rows use an effective bitwidth of 2.75 at INT2; QVG rows on MAGI-1 use the default QVG configuration, which yields an effective bitwidth of approximately 2.52. Standard errors for all metrics are reported in Tables 5 and 9.

Model	Quantization scheme	Precision	With correction	PSNR↑	SSIM↑	LPIPS↓	VBench↑
MAGI-1	—	BF16		–	–	–	78.27
	RTN	INT2	×	23.17	0.799	0.195	76.67
			✓	24.08	0.828	0.131	77.86
	QuaRot+RTN	INT2	×	17.10	0.630	0.453	70.24
			✓	22.97	0.801	0.165	78.02
	QVG	INT2	×	23.01	0.826	0.132	77.81
			✓	**25.29**	**0.856**	**0.107**	**78.23**
SkyReels-V2	—	BF16		–	–	–	78.89
	RTN	INT2	×	18.38	0.670	0.383	68.83
			✓	18.39	0.743	0.269	**80.04**
	QuaRot+RTN	INT2	×	19.20	0.708	0.319	71.44
			✓	**20.42**	**0.784**	**0.202**	78.58
HY-WorldPlay	RTN	INT2	×	16.96	0.573	0.390	–
			✓	17.15	0.577	0.311	–
	QuaRot+RTN	INT2	×	17.16	0.575	0.376	–
			✓	**18.27**	**0.616**	**0.273**	–

5 EXPERIMENTS

We evaluate the correction end to end on all three models using video-quality metrics averaged over prompts (Section 5.2). On MAGI-1, we additionally study attention statistics, key versus value quantization, and the memory–quality trade-off (Section 5.3).

5.1 EXPERIMENTAL SETUP

Models. Our experiments cover three autoregressive video diffusion models: MAGI-1 (4.5B; Sand.ai et al., 2025), SkyReels-V2 (1.3B; Chen et al., 2025), and HY-WorldPlay (8B; Sun et al., 2025). All use chunk-wise generation with KV caching over previously generated chunks. MAGI-1 uses 16 denoising steps with a sliding window annealed from 5 to 2 chunks, SkyReels-V2 uses 50 steps with a 5-chunk window, and HY-WorldPlay uses 4 steps. Unless otherwise noted, all other generation hyperparameters remain at default values.

Quantization configuration. We adopt group-wise per-token asymmetric INT2 quantization of key and value states as the default KV-cache compression setting throughout the paper. Unless otherwise noted, we use group size $g = 32$, FP8 E4M3 scales, and BF16 zero-points. We evaluate two quantization schemes: QuaRot+RTN (Ashkboos et al., 2024) and plain RTN without rotation. Additionally, on MAGI-1 we include QuantVideoGen (QVG; Xi et al., 2026) using its default configuration (S=1, B=64, K=256 in its own notation) to demonstrate that our correction composes with upstream video-aware cache compression. We apply the Taylor-approximated bias correction from Section 4.2 to all quantization schemes. Unquantized BF16 results serve as the reference outputs for fidelity metrics.

Metrics. We report the fidelity metrics PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018) to measure the similarity between quantized and BF16 outputs on identical inputs. We further evaluate generated videos using the VBench evaluation framework (Huang et al., 2023) in the VBench-Long setting from VBench++ (Huang et al., 2024), which adapts the benchmark to long-form

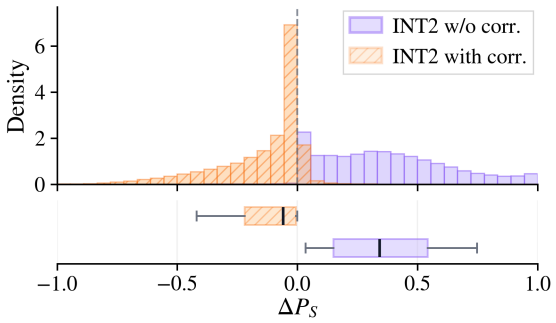


Figure 4: Shift in attention mass assigned to the cached block of tokens before (purple) and after (orange) our correction on MAGI-1 under INT2 QuaRot+RTN. Positive values indicate that the quantized cached tokens steal attention from the current unquantized chunk. The median bias is large under INT2 quantization, and our correction significantly reduces this bias toward zero.

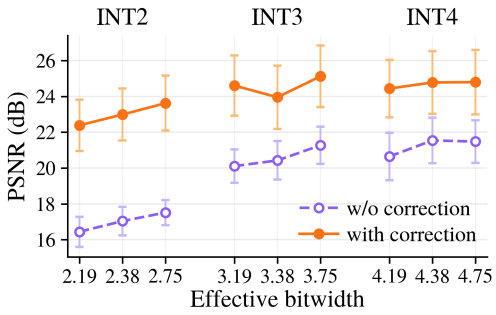


Figure 5: Trade-off between image quality, measured by PSNR, and memory footprint of the KV cache, measured as effective bitwidth per element, on MAGI-1 under QuaRot+RTN quantization. Bitwidths correspond to group sizes $g \in \{128, 64, 32\}$. The sweep uses a reduced prompt set. Whiskers indicate standard error.

videos. Table 1 reports the aggregate VBench score; per-dimension results and Quality/Semantic sub-scores are provided in Appendix D.3.

Evaluation data. For MAGI-1 and SkyReels-V2, we evaluate on the first 30% of prompts from each VBench-Long dimension, generating 10-second videos (240 frames) and 7-second videos (177 frames), respectively. This subset was selected using a fixed rule defined before inspecting any results. We do not evaluate on the full prompt set, as this is computationally prohibitive across all models and quantization configurations. All comparisons are paired on identical prompts and seeds; Appendix D.2 reports paired significance tests and per-prompt win rates. For HY-WorldPlay, we generate 10-second videos (253 frames) from the 10 image–prompt pairs released in the official repository (Sun et al., 2025). We do not report VBench scores for this model, as its required inputs (image, text prompt, and per-frame keyboard actions) are not provided by any VBench suite. HY-WorldPlay therefore serves as supporting evidence on a third, action-conditioned architecture, while the broader benchmark evidence comes from MAGI-1 and SkyReels-V2.

5.2 Main Results

KV-cache quantization substantially degrades video quality (Table 1, Fig. 1, Figs. 8 and 9). Our correction improves every fidelity metric (PSNR, SSIM, LPIPS) across all three models and all quantization schemes, without any model-specific tuning (Table 1). On MAGI-1 and SkyReels-V2, it also improves the VBench score and closes the quality gap between INT2 KV-cache quantization and the BF16 baseline. On SkyReels-V2 with RTN, the corrected score slightly exceeds BF16 (80.04 vs. 78.89), a difference of only about 1.2 combined standard errors (Table 9). The MAGI-1 per-dimension breakdown in Appendix D.3 shows that these gains are broad-based across the VBench dimensions. On HY-WorldPlay, where VBench is not applicable, the fidelity gains are clearest under QuaRot+RTN.

Paired tests on the per-prompt results (Appendix D.2) find 18 of the 21 fidelity improvements statistically significant; the remaining three (SkyReels-V2 RTN PSNR, HY-WorldPlay RTN PSNR and SSIM) are within standard error. Finally, composed with QVG, our correction achieves the best quantized MAGI-1 results on all metrics, confirming that the two methods are complementary

5.3 Ablations and Analysis

All results in this section use MAGI-1 with QuaRot+RTN quantization, which scores higher on VBench under our correction than plain RTN (Table 1).

Attention-level effects. Attention stealing caused by the Jensen bias is illustrated in Fig. 2. We quantify this effect by measuring the shift in attention mass assigned to cached tokens, $\Delta P_{\mathcal{S}} = \hat{P}_{\mathcal{S}} - P_{\mathcal{S}}$, aggregated across all layers, denoising steps, and attention heads. Figure 4 shows that under INT2 quantization, $\Delta P_{\mathcal{S}}$ is strongly positive, confirming that cached tokens steal attention mass. Our correction shifts the distribution back toward zero, though it slightly over-corrects into negative values, consistent with the calibration in Appendix B, where b exceeds the measured bias by 8% under QuaRot+RTN. The same holds further along the attention pipeline: the correction also reduces the Jensen–Shannon divergence (JSD) between quantized and BF16 attention weights (Appendix E.2) and the attention-output MSE (Appendix E.3), with the three improvements occurring mostly on the same heads (Appendix E.4). Corresponding INT4 results are provided in Appendices E.1 to E.3.

Key versus value quantization. Key-only and value-only ablations show that the degradation stems almost entirely from key quantization, the path our correction addresses. Under key-only quantization, the correction recovers $\sim 96\%$ of the VBench gap to BF16 (Appendix D.4).

Memory–quality trade-off. With our correction, INT2 at 2.19 effective bits matches or exceeds uncorrected INT4 at 4.38 bits in PSNR (Fig. 5), SSIM, and LPIPS (Appendix D.5) while halving the memory. More broadly, the correction improves all three metrics at every tested bitwidth and group size, shifting the entire memory–quality trade-off toward higher quality.

5.4 Cross-Domain Experiment: LLM Chunked Prefill

Although our main experiments target chunk-wise video diffusion, chunked LLM prefill has a similar cached/current attention structure: a quantized cached prefix and a multi-token current prefill block appear in the same softmax. We run a small-scale diagnostic study on three decoder-only LLMs using LongBench-Pro English prompts (Chen et al., 2026b). We compare BF16, INT2, and INT2 with our Taylor correction on paired configurations using teacher-forced negative log-likelihood (NLL).

Across the LLM experiments, INT2 generally increases NLL relative to BF16, while the Taylor correction reduces NLL relative to plain INT2. This is consistent with the mechanism studied in our video experiments, but we do not interpret it as a comprehensive LLM benchmark. Details and prompt-length breakdowns are provided in Appendix F.

6 Discussion and Conclusion

We identify the Jensen bias, by which zero-mean key quantization noise inflates the softmax contribution of cached keys, and derive a per-attention-score correction that removes it in expectation. Its second-order Taylor approximation adds an estimated 1.8% end-to-end latency on MAGI-1 and no memory (Section 4.3), consistently improves fidelity across MAGI-1, SkyReels-V2, and HY-WorldPlay, and yields large VBench gains on MAGI-1 and SkyReels-V2. Because our method acts only on attention scores, it composes with upstream quantizers; combined with QuantVideoGen, it achieves the best quantized MAGI-1 results (Table 1).

Limitations & future work. The attention-mass shift arises whenever quantized cached keys and unquantized current keys share a softmax. Chunk-wise video diffusion, with thousands of current tokens per chunk, is a prominent instance and the focus of our experiments. Our preliminary LLM results show that the correction also reduces NLL under chunked prefill with KV-cache quantization (Gokhale et al., 2025), which merits a full-scale evaluation. Single-token decoding, with one unquantized current token against many cached ones, offers less headroom. The Taylor correction relies on an approximately zero-mean, uniform quantization-noise model and removes the bias only in expectation. Under plain RTN it overestimates the bias by roughly 30% (8% under QuaRot+RTN, Appendix B), and its gain is limited when attention concentrates on a few cached tokens. Value noise enters linearly after the softmax and thus causes no Jensen bias; it is outside the scope of our correction and small in our setting (Table 10). Our VBench evaluation uses a prompt subset, so our claims concern paired effects on identical inputs (Appendix D.2). For floating-point formats (FP, MXFP, NVFP), whose non-uniform grids produce a different noise distribution, we derive the variance substitution (Appendix A.2) but leave empirical evaluation open.

AI USE STATEMENT

In this work, we used generative AI tools in three ways: to aid and polish writing (drafting and revising manuscript text, consolidating material from an earlier version of this manuscript, shortening sections), for retrieval and discovery (suggesting candidates for related work), and for research ideation and execution (aiding in discussing the derivation, suggesting ablations to run, helping write boilerplate experiment code, and analyzing results).

All AI use was directed by the authors, and we verified every AI-assisted contribution: we read and gave final edits to every AI-assisted passage, read the cited work in the original before referencing it, checked the derivation in Section 4 and Appendix A step by step, and reviewed and tested all AI-assisted code. We take full responsibility for the content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work proposes a training-free correction for KV-cache quantization in autoregressive video diffusion models. The direct goal is to improve the efficiency and quality of long-form video generation by reducing memory usage while preserving generation fidelity. Potential positive impacts include lowering the computational cost of research on long-video and world-model generation, enabling longer context windows under fixed memory budgets, and improving accessibility of efficient inference methods for academic and resource-constrained settings.

REPRODUCIBILITY STATEMENT

The method and its assumptions are described in Section 4, with extensions in Appendix A and empirical validation of the noise model in Appendix B. Experimental settings are given in Section 5.1; implementation and runtime details appear in Appendices C.1 and C.4. Additional results, standard errors, and paired significance tests are provided in Appendices D.1 to D.3, and the LLM chunked-prefill protocol is described in Appendix F.1. Appendix C.2 gives pseudocode for the corrected attention, and Appendix C.3 describes how the correction is integrated into attention. All evaluated models and benchmarks are public. Section 5.1 reports the model sizes, generation settings, quantization configuration, and the prompt-selection rule, and all comparisons use identical prompts and seeds. We do not release code with this submission. The correction itself is given in closed form (Section 4.2) and as pseudocode (Appendix C.2), so it can be reimplemented from the paper. Reproducing our exact numbers additionally requires the evaluation scripts and fused kernels, which we will release.

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.

Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion Forcing: Next-token prediction meets full-sequence diffusion, 2024. URL https://arxiv.org/abs/2407.01392.

Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, Weiming Xiong, Wei Wang, Nuo Pang, Kang Kang, Zhiheng Xu, Yuzhe Jin, Yupeng Liang, Yubing Song, Peng Zhao, Boyuan Xu, Di Qiu, Debang Li, Zhengcong Fei, Yang Li, and Yahui Zhou. SkyReels-V2: Infinite-length film generative model, 2025. URL https://arxiv.org/abs/2504.13074.

Shuo Chen, Cong Wei, Sun Sun, Ping Nie, Kai Zhou, Ge Zhang, Ming-Hsuan Yang, and Wenhu Chen. Context Forcing: Consistent autoregressive video generation with long context, 2026a. URL https://arxiv.org/abs/2602.06028.

Ziyang Chen, Xing Wu, Junlong Jia, Chaochen Gao, Qi Fu, Debing Zhang, and Songlin Hu. LongBench Pro: A more realistic and comprehensive bilingual long-context evaluation benchmark, 2026b. URL https://arxiv.org/abs/2601.02872.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning, 2023. URL https://arxiv.org/abs/2307.08691.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale, 2022. URL https://arxiv.org/abs/2208.07339.

Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. Flex Attention: A programming model for generating optimized attention kernels, 2024. URL https://arxiv.org/abs/2412.05496.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Xiang Gao, Meera Sitharam, and Adrian E. Roitberg. Bounds on the Jensen gap, and implications for mean-concentrated distributions. *arXiv preprint arXiv:1712.05267*, 2020. doi: 10.48550/arXiv.1712.05267. URL https://arxiv.org/abs/1712.05267.

Sai Gokhale, Devleena Das, Rajeev Patwari, Ashish Sirasao, and Elliott Delaye. KV Pareto: Systems-level optimization of KV cache and model compression for long context inference, 2025. URL https://arxiv.org/abs/2512.01953.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. *arXiv preprint arXiv:2401.18079*, 2024.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models, 2023. URL https://arxiv.org/abs/2311.17982.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench++: Comprehensive and versatile benchmark suite for video generative models, 2024. URL https://arxiv.org/abs/2411.13503.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023. URL https://arxiv.org/abs/2310.06825.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. HunyuanVideo: A systematic framework for large video generative models, 2025. URL https://arxiv.org/abs/2412.03603.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. *arXiv preprint arXiv:2309.06180*, 2023.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=L057s2Rq8O.

Chengtao Lv, Yumeng Shi, Yushi Huang, Ruihao Gong, Shen Ren, and Wenya Wang. Light Forcing: Accelerating autoregressive video diffusion via sparse attention, 2026. URL https://arxiv.org/abs/2602.04789.

Yuexiao Ma, Xuzhe Zheng, Jing Xu, Xiwei Xu, Feng Ling, Xiawu Zheng, Huafeng Kuang, Huixia Li, Xing Wang, Xuefeng Xiao, Fei Chao, and Rongrong Ji. Flow caching for autoregressive video generation, 2026. URL https://arxiv.org/abs/2602.10825.

Meta. Meta Llama 3.1 8B model card. https://huggingface.co/meta-llama/Llama-3.1-8B, 2024. Accessed: 2026-05-07.

Mistral AI. Mistral-7B-Instruct-v0.3 model card. https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3, 2024. Accessed: 2026-05-07.

NVIDIA, Felix Abecassis, Anjulie Agrusa, Dong Ahn, Jonah Alben, Stefania Alborghetti, Michael Andersch, Sivakumar Arayandi, Alexis Bjorlin, Aaron Blakeman, Evan Briones, Ian Buck, Bryan Catanzaro, Muya Chang, Jinhang Choi, Mike Chrzanowski, Eric Chung, Victor Cui, Steve Dai, Bita Darvish Rouhani, Carlo del Mundo, Deena Donia, Burc Eryilmaz, Henry Estela, Abhinav Goel, Oleg Goncharov, Yugi Guvvala, Robert Hesse, Russell Hewett, Herbert Hum, Ujval Kapasi, Brucek Khailany, Mikail Khona, Nick Knight, Alex Kondratenko, Ronny Krashinsky, Ben Lanir, Simon Layton, Michael Lightstone, Daniel Lo, Paulius Micikevicius, Asit Mishra, Tim Moon, Deepak Narayanan, Chao Ni, Abhijit Paithankar, Satish Pasumarthi, Ankit Patel, Mostofa Patwary, Ashwin Poojary, Gargi Prasad, Sweta Priyadarshi, Yigong Qin, Xiaowei Ren, Oleg Rybakov, Charbel Sakr, Sanjeev Satheesh, Stas Sergienko, Pasha Shamis, Kirthi Shankar, Nishant Sharma, Mohammad Shoeybi, Michael Siu, Misha Smelyanskiy, Darko Stosic, Dusan Stosic, Bor-Yiing Su, Frank Sun, Nima Tajbakhsh, Shelby Thomas, Przemek Tredak, Evgeny Tsykunov, Gandhi Vaithilingam, Aditya Vavre, Rangharajan Venkatesan, Roger Waleffe, Qiyu Wan, Hexin Wang, Mengdi Wang, Lizzie Wei, Hao Wu, Evan Wu, Keith Wyss, Ning Xu, Jinze Xue, Charlene Yang, Yujia Zhai, Ruoxi Zhang, Jingyang Zhu, and Zhongbo Zhu. Pretraining large language models with NVFP4, 2026. URL https://arxiv.org/abs/2509.25149.

Nilesh Prasad Pandey, Marios Fournarakis, Chirag Patel, and Markus Nagel. Softmax bias correction for quantized generative models, 2023. URL https://arxiv.org/abs/2309.01729.

Qwen. Qwen2.5-32B-Instruct model card. https://huggingface.co/Qwen/Qwen2.5-32B-Instruct, 2024. Accessed: 2026-05-07.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Bita Darvish Rouhani et al. OCP microscaling formats (MX) specification version 1.0. Technical report, Open Compute Project, 2023. URL https://www.opencompute.org/documents/ocp-microscaling-formats-mx-v1-0-spec-final-pdf.

Dvir Samuel, Issar Tzachor, Matan Levy, Michael Green, Gal Chechik, and Rami Ben-Ari. Fast autoregressive video diffusion and world models with temporal cache compression and sparse attention, 2026. URL https://arxiv.org/abs/2602.01801.

Sand.ai, Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, W. Q. Zhang, Weifeng Luo, Xiaoyang Kang, Yuchen Sun, Yue Cao, Yunpeng Huang, Yutong Lin, Yuxin Fang, Zewei Tao, Zheng Zhang, Zhongshu Wang, Zixun Liu, Dai Shi, Guoli Su, Hanwen Sun, Hong Pan, Jie Wang, Jiexin Sheng, Min Cui, Min Hu, Ming Yan, Shucheng Yin, Siran Zhang, Tingting Liu, Xianping Yin, Xiaoyu Yang, Xin Song, Xuan Hu, Yankai Zhang, and Yuqiao Li. MAGI-1: Autoregressive video generation at scale, 2025. URL https://arxiv.org/abs/2505.13211.

Utkarsh Saxena and Kaushik Roy. KVLinC : KV cache quantization with Hadamard rotation and linear correction, 2025. URL https://arxiv.org/abs/2510.05373.

Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. WorldPlay: Towards long-term geometric consistency for real-time interactive world modeling, 2025. URL https://arxiv.org/abs/2512.14614.

Qian Tao, Wenyuan Yu, and Jingren Zhou. AsymKV: Enabling 1-bit quantization of KV cache with layer-wise asymmetric quantization configurations, 2024. URL https://arxiv.org/abs/2410.13212.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025. URL https://arxiv.org/abs/2503.20314.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

B. Widrow, I. Kollár, and Ming-Chang Liu. Statistical theory of quantization. *IEEE Transactions on Instrumentation and Measurement*, 45(2):353–361, 1996. doi: 10.1109/19.492748.

Haocheng Xi, Shuo Yang, Yilong Zhao, Muyang Li, Han Cai, Xingyang Li, Yujun Lin, Zhuoyang Zhang, Jintao Zhang, Xiuyu Li, Zhiying Xu, Jun Wu, Chenfeng Xu, Ion Stoica, Song Han, and Kurt Keutzer. Quant VideoGen: Auto-regressive long video generation via 2-bit KV-cache quantization, 2026. URL https://arxiv.org/abs/2602.02958.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-video diffusion models with an expert transformer, 2025. URL https://arxiv.org/abs/2408.06072.

Yuzhe Yao, Feng Tian, Jun Chen, Haonan Lin, Guang Dai, Yong Liu, and Jingdong Wang. Timestep-aware correction for quantized diffusion models, 2024. URL https://arxiv.org/abs/2407.03917.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models, 2025. URL https://arxiv.org/abs/2412.07772.

Amir Zandieh, Majid Daliri, Majid Hadian, and Vahab Mirrokni. TurboQuant: Online vector quantization with near-optimal distortion rate, 2025. URL https://arxiv.org/abs/2504.19874.

Jintao Zhang, Haofeng Huang, Pengle Zhang, Jia Wei, Jun Zhu, and Jianfei Chen. SageAttention2: Efficient attention with thorough outlier smoothing and per-thread INT4 quantization. In *International Conference on Machine Learning (ICML)*, 2025a.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. URL https://arxiv.org/abs/1801.03924.

Tao Zhang, Ziqian Zeng, Hao Peng, Huiping Zhuang, and Cen Chen. MixKVQ: Query-aware mixed-precision KV cache quantization for long-context reasoning, 2025b. URL https://arxiv.org/abs/2512.19206.

A Extensions of the Correction

A.1 Extension to QuaRot

The derivation in Section 4 is carried out in the unrotated space. Section 4.2 states that the correction extends to QuaRot (Section 3) by replacing q with the rotated query Hq. Here we show why this replacement is valid. With the Hadamard matrix H applied to both keys and queries, the quantized score becomes

$$\hat{s}_i = \frac{(Hq)^\top (Hk_i + \epsilon_i)}{\sqrt{d}} = s_i + \delta_i^{(H)}, \tag{11}$$

where ϵ_i now denotes the quantization error of the rotated key Hk_i, the term $(Hq)^\top Hk_i/\sqrt{d}$ equals s_i because H is orthogonal, and $\delta_i^{(H)} = (Hq)^\top \epsilon_i/\sqrt{d}$. The key error ϵ_i satisfies the same noise model as in steps (ii) and (iii) of Section 4.2, with step sizes $\Delta_{i,j}$ taken from the rotated key. The noise $\delta_i^{(H)}$ thus has the same form as δ_i in Section 4.1, with q replaced by Hq, and the derivation applies without change. In Eq. (10), $\|q_j\|^2$ becomes $\|(Hq)_j\|^2$, which generally differs from $\|q_j\|^2$ because the rotation mixes channels across groups. The correction must therefore use the rotated query.

A.2 Extension to Floating-Point and Mixed-Precision Formats

The derivation in Section 4.2 requires only that the per-element round-trip error be zero-mean and uniform on $\pm\Delta_{i,c}/2$ and independent across channels. The step size $\Delta_{i,c}$ is the sole format-dependent quantity; it enters quadratically in Eq. (10).

Floating-point formats. Floating-point grids are uniform within a binade, with spacing $\Delta = 2^{E-m}$ (multiplied by the block scale for block formats such as MXFP and NVFP), where E is the element's exponent and m the number of mantissa bits. Round-to-nearest error is therefore uniform on $\pm\Delta/2$ within a binade, exactly as in the integer case, and the correction applies unchanged with a per-element $\Delta_{i,c}$. The spacing is recoverable from the stored element's own exponent field, so the correction still needs no metadata beyond what dequantization already reads. One caveat, absent in the integer case, is that uniformity holds only within a binade: elements near a binade boundary see a mixture of two spacings whose error is not exactly zero-mean.

The error structure does differ. Integer quantization gives approximately constant absolute error within a group, whereas floating point gives approximately constant relative error, so the spacing is coarsest on the largest-magnitude elements. At equal nominal bitwidth, FP4 (E2M1) has a step size on top-binade channels several times larger than INT4. The Jensen bias is a $q_c^2\Delta_{i,c}^2$-weighted sum that emphasizes exactly these channels, and attention scores are dominated by large-magnitude key channels. We therefore expect the bias to be no smaller for floating-point than for integer formats at equal bitwidth, in which case the correction would matter at least as much. Confirming this requires floating-point experiments that we have not run.

Scaling with bitwidth. The Taylor term is quadratic in the step size and Δ shrinks geometrically with bitwidth, so the Jensen bias falls off rapidly as precision increases. Our INT4 results are consistent with this: the attention-mass shift (Appendix E.1) and the reduction in attention-weight JSD (Appendix E.2) are markedly smaller at INT4 than at INT2. The diminishing benefit at higher bitwidth is thus a consequence of the error model. Floating-point formats below 4 bits are rare in both the literature and hardware, so the regime where degradation is most severe, and the correction matters most, is low-bit integer quantization, which is what we evaluate.

Mixed precision. Mixed-precision caches require no special handling. The correction is computed from per-element step sizes $\Delta_{i,c}$, which carry no format tag: the bitwidth may vary across layers, heads, tokens, or channels, and integer and floating-point elements may be mixed within a single cache. Channels kept at higher precision have small $\Delta_{i,c}$ and contribute correspondingly little, so the correction is dominated by the low-precision channels. Query-aware mixed-precision schemes such as MixKVQ (Zhang et al., 2025b) are therefore compatible by construction.

B Empirical Validation of the Quantization-Noise Model

We test the three steps of Section 4.2 on MAGI-1 with group-wise per-token asymmetric INT2 quantization (g=32) under both RTN and QuaRot+RTN, on the same prompts as the attention-level statistics of Section 5.3.

Calibration protocol. For each attention head, denoising step, layer, and query chunk, we consider every pair of a query and a cached key. For each pair, we compute the predicted correction b from Eq. (10) and the realized score noise $\delta = q^\top(\hat{k} - k)/\sqrt{d}$. A single pair yields one realization of δ, whereas the Jensen bias $y = \log \mathbb{E}[e^\delta]$ is an expectation. We therefore group pairs by their predicted correction into 30 bins with $\sqrt{2}$-spaced edges covering $b \in [2^{-3}, 2^{12}]$, pool all pairs of a bin across heads, layers, steps, and prompts, and estimate $\mathbb{E}[\delta]$, $\mathrm{Var}(\delta)$, and y within each bin. Using the second-order expansion $y \approx \mathbb{E}[\delta] + \frac{1}{2}\mathrm{Var}(\delta)$ and dividing by $b = \sigma^2_{\mathrm{pred}}/2$, where σ^2_{pred} is the variance predicted by Eq. (9), gives

$$\frac{y}{b} \approx \frac{\mathbb{E}[\delta]}{b} + \frac{\mathrm{Var}(\delta)}{\sigma^2_{\mathrm{pred}}}, \tag{12}$$

which motivates the three quantities reported in Table 2: $\mathrm{Var}(\delta)/\sigma^2_{\mathrm{pred}}$ tests the variance prediction (step iii), $\mathbb{E}[\delta]/b$ measures the contribution of any mean shift (step ii), and y/b compares the predicted correction with the measured bias, testing all three steps jointly. Their ideal values are 1, 0, and 1. We report pair-count-weighted averages over bins in Table 2.

Table 2: Calibration of the three approximation steps of Section 4.2 on MAGI-1 at INT2 (g=32). Pair-count-weighted averages over bins of predicted correction b; σ^2_{pred} is the variance predicted in step (iii) and $y = \log \mathbb{E}[e^\delta]$ the measured Jensen bias.

Quantity	Tests	Ideal	RTN	QuaRot+RTN
$\mathrm{Var}(\delta)/\sigma^2_{\mathrm{pred}}$	step (iii), variance prediction	1	1.284	0.978
$\mathbb{E}[\delta]/b$	step (ii), zero mean	0	-0.168	-0.014
y/b	all three steps	1	0.772	0.928

Results. QuaRot+RTN closely matches all three ideals: predicted and measured variance agree to about 2%, the mean term is negligible, and the correction tracks the observed Jensen bias ($y/b = 0.928$). RTN deviates more. Its measured variance exceeds the prediction by 28% and its score-error mean is negative, so by Eq. (12) the first two cumulants predict $y/b \approx 1.12$, whereas the measured value is 0.772. The mean and variance therefore do not explain the overshoot; the remaining gap lies in the cumulants beyond the variance that step (i) truncates. The variance mismatch under RTN is concentrated in three outlier layers with extreme attention stealing, a behavior not observed under QuaRot+RTN. Even so, $y/b = 0.772$ captures the correct sign and scale. Uncorrected, the bias is $0.772\,b$; subtracting b leaves a residual of $0.228\,b$, so the correction removes 70% of the bias, a $3.4\times$ reduction, and remains accurate enough to produce consistent downstream improvements (Table 1). The correction is thus effective under both schemes and most accurate under QuaRot+RTN.

Key-error and score-noise moments. As a complementary check, we examine the key-space quantization error $\epsilon = \hat{K} - K$ and the induced score noise δ directly. For each attention head, we compare both errors with their ideal distributions. We divide each key error by the step size of its group, $\epsilon_{i,c}/\Delta_{i,j(c)}$, so that it should follow $\mathcal{U}(-\frac{1}{2}, \frac{1}{2})$ in every group, and compute its mean, variance, skew, excess kurtosis, and saturation rate. Similarly, for the score noise, whose mean and variance Table 2 already tests, we compute the skew and excess kurtosis, which should match a Gaussian. Table 3 reports these statistics averaged across prompts, layers, denoising steps, chunks, and heads. "True saturation" counts pre-clamp quantization codes that fall outside the representable INT2 range; the small non-zero rate arises because the per-group scales are stored in FP8, whose rounding slightly perturbs the range covered by each group.

The key errors are approximately zero-mean and closely follow the uniform model under both schemes, with negligible saturation. On average across heads, the induced score noise is nearly

Table 3: Moments of the normalized key quantization error and of the induced score noise on MAGI-1 at INT2 (g=32), averaged over prompts, layers, denoising steps, chunks, and heads.

Space	Statistic	Ideal	RTN	QuaRot+RTN
Key error ϵ/Δ	Mean	0	0.0007	-0.0020
	$12\,\mathrm{Var}(\epsilon/\Delta)$	1	0.945	1.016
	Skew	0	-0.002	0.013
	Excess kurtosis	-1.2	-1.082	-1.196
	True saturation	0%	0.022%	0.015%
Score noise δ	Skew	0	0.005	0.008
	Excess kurtosis	0	0.045	0.013

symmetric and Gaussian-like, which supports truncating the cumulant expansion after the variance term (step i). The RTN outlier layers discussed above are the exception. QuaRot+RTN matches the predicted variance and kurtosis more closely and saturates less, consistent with the Hadamard rotation spreading channel-wise outliers before quantization. The correction is therefore not specific to one quantizer: it remains effective with plain RTN, while quantizers whose errors follow the assumed model more closely allow a more accurate correction.

C Cost, Implementation, and Runtime

C.1 Cost Breakdown

We detail the per-query, per-key, per-score-entry and total costs for the Taylor correction under group-wise per-token quantization with $G = d/g$ groups, where Q and K denote the numbers of queries and cached keys and $\mathcal{C}_j$ denotes the channels of group j.

- **Per query:** Compute $\|q_j\|^2 = \sum_{c\in\mathcal{C}_j} q_c^2$ for each group $j = 1, \ldots, G$, costing $O(d)$.
- **Per key:** Compute $\Delta_{i,j}^2/(24d)$ for each group, costing $O(G)$.
- **Per score entry:** Compute the inner product of the per-query vector $(\|q_j\|^2)_{j=1}^G$ and the per-key vector $(\Delta_{i,j}^2/(24d))_{j=1}^G$, costing $O(G)$.
- **Total:**
$$O(Q \cdot d + K \cdot G + Q \cdot K \cdot G). \tag{13}$$
Since $G = d/g$ and $K \gg d$, the dominant term is $O(Q \cdot K \cdot d/g)$. Compared to the attention cost $O(Q \cdot K \cdot d)$, this is lower by a factor of g.

On storage, we note that a cached key of dimension d quantized to B bits per element with group size g requires $d \cdot B$ bits for the quantized values, plus metadata per group: one scale stored in FP8 E4M3 (8 bits) and one zero-point stored in BF16 (16 bits), for a total of 24 bits per group. With $G = d/g$ groups per token, the effective bitwidth is

$$B_{\text{eff}} = \frac{d \cdot B + 24 \cdot G}{d} = B + \frac{24}{g}. \tag{14}$$

Our correction adds no storage beyond this ($\Delta_{i,j}$ is the scale itself). For our default configuration ($d = 128$, $g = 32$), this yields $B_{\text{eff}} = 2.75$ at INT2.

C.2 Pseudocode for Taylor-Corrected Attention

Algorithm 1 provides the full pseudocode for attention with the Taylor correction applied to quantized cached keys, as derived in Section 4.2.

C.3 Implementation

We apply the correction on the fly through a `score_mod` function in PyTorch's FlexAttention (Dong et al., 2024), which subtracts b_i from each cached score inside the fused attention kernel and thus

Algorithm 1: Attention with Taylor correction for quantized cached keys (group-wise)

Input: Query matrix $Q \in \mathbb{R}^{M \times d}$; cached quantized keys $K^q_{\mathcal{S}}$ with per-group step sizes $\{\Delta_{i,j}\}$ and zero-points $\{z_{i,j}\}$; cached values $V_{\mathcal{S}}$; current-chunk keys $K_{\mathcal{R}}$; current-chunk values $V_{\mathcal{R}}$; group size g, number of groups $G = d/g$; value dimension d_v
Output: Attention output $O \in \mathbb{R}^{M \times d_v}$

$\hat{K}_{\mathcal{S}} \leftarrow \mathrm{dequant}(K^q_{\mathcal{S}})$;
$S_{\mathcal{S}} \leftarrow Q\hat{K}_{\mathcal{S}}^{\top}/\sqrt{d}$;
$S_{\mathcal{R}} \leftarrow QK_{\mathcal{R}}^{\top}/\sqrt{d}$;
for $m = 1$ **to** M **do**
  for $j = 1$ **to** G **do**
    $\nu_{m,j} \leftarrow \sum_{c \in \mathcal{C}_j} Q^2_{m,c}$;
  end
  forall $i \in \mathcal{S}$ **do**
    $b_{m,i} \leftarrow \dfrac{1}{24\,d} \sum_{j=1}^{G} \Delta^2_{i,j}\, \nu_{m,j}$;
    $S_{\mathcal{S}}[m,i] \leftarrow S_{\mathcal{S}}[m,i] - b_{m,i}$;
  end
end
$S \leftarrow \mathrm{concat}(S_{\mathcal{S}}, S_{\mathcal{R}})$;
$P \leftarrow \mathrm{softmax}(S)$;
$V \leftarrow \mathrm{concat}(V_{\mathcal{S}}, V_{\mathcal{R}})$;
$O \leftarrow PV$;
return O

avoids materializing a correction tensor the size of the full score matrix. The runtime benchmarks instead use fused Triton kernels (Appendix C.4).

All MAGI-1 generation experiments were conducted on NVIDIA L4 GPUs, SkyReels-V2 experiments on NVIDIA A100 GPUs, and HY-WorldPlay experiments on NVIDIA A100 80 GB GPUs.

C.4 Runtime Benchmarks

Our FlexAttention implementation (Appendix C.3) was designed for rapid experimentation and functional validation rather than kernel-level optimization, so its latency does not provide a representative measure of the correction's practical cost. We therefore implemented two fused Triton attention kernels for NVIDIA A100: a base kernel and an otherwise identical correction kernel that computes and subtracts our Taylor correction (Eq. (10)) on the fly. We compare both against FlashAttention-2 (FA2; Dao, 2023). Comparing the base and correction Triton kernels isolates the overhead of our correction, while comparison against FA2 reflects the performance relative to a commonly used attention backend.

We benchmark MAGI-1 attention shapes over all context lengths ($K = m \times 12{,}150$ keys for $m \in \{1, \ldots, 5\}$) and cached-key fractions $\rho \in \{0.10, 0.25, 0.33, 0.50, 0.75\}$. Each entry in Table 4 is the median of 120 executions.

Table 4: Latency per attention call on NVIDIA A100 in milliseconds, given as FA2 / Triton base / Triton with correction, for MAGI-1 attention shapes with K keys of which a fraction ρ is cached. Medians over 120 executions.

$\rho \setminus K$	12,150	24,300	36,450	48,600	60,750
0.10	8.6 / 9.6 / 10.4	17.1 / 19.1 / 20.6	25.8 / 28.8 / 30.6	34.4 / 38.5 / 41.7	43.1 / 48.1 / 50.3
0.25	8.6 / 9.7 / 10.4	17.1 / 19.1 / 20.6	25.8 / 28.8 / 30.7	34.4 / 38.4 / 41.6	43.2 / 48.3 / 50.7
0.33	8.6 / 9.7 / 10.4	17.1 / 19.2 / 20.6	25.8 / 28.8 / 30.7	34.4 / 38.4 / 41.7	43.2 / 48.3 / 50.8
0.50	8.6 / 9.7 / 10.4	17.2 / 19.3 / 20.5	25.8 / 28.8 / 30.9	34.4 / 38.4 / 41.6	43.2 / 48.3 / 51.0
0.75	8.6 / 9.7 / 10.5	17.1 / 19.2 / 20.6	25.8 / 28.8 / 31.1	34.4 / 38.4 / 41.7	43.1 / 48.2 / 51.3

Across all 25 configurations, the base Triton implementation is 11.8% slower than FA2 on average, showing that it remains reasonably close to the highly optimized baseline. More importantly, comparing the two matched Triton kernels, our correction adds only 7.2% latency on average (4.6–8.5% across configurations).

Estimated end-to-end impact. The correction affects only the attention computation. If attention accounts for a fraction f of total generation latency, the estimated end-to-end overhead of the correction is $0.072\,f$ relative to a matched baseline. In our experiments attention accounts for $\sim$25% of total generation latency, so the correction adds $\sim$1.8% end-to-end latency relative to the matched Triton baseline and $\sim$4.8% relative to FA2, of which $\sim$3% is attributable to the Triton backend itself. The share f depends on the model and on how efficiently the remaining components are implemented, so we report the kernel-level overhead as the primary number. Importantly, peak GPU memory is unchanged: the correction stores no additional state, so the full INT2 memory saving is preserved.

D Additional End-to-End Results

D.1 Fidelity Metrics with Standard Errors

Table 1 reports fidelity metrics (PSNR, SSIM, LPIPS) averaged across prompts. Table 5 reports the same values with standard errors computed across prompts (the independent sampling unit), using the evaluation data described in Section 5.1.

Table 5: Fidelity metrics with standard errors for all configurations in Table 1. PSNR, SSIM, and LPIPS are computed relative to the BF16 reference; $\pm$ denotes standard error across prompts. Best quantized result per model is **bolded**.

Model	Quant. scheme	Prec.	With corr.	PSNR↑	SSIM↑	LPIPS↓
MAGI-1	RTN	INT2	×	23.17 ± 0.24	0.799 ± 0.006	0.195 ± 0.006
			✓	24.08 ± 0.30	0.828 ± 0.006	0.131 ± 0.004
	QuaRot+RTN	INT2	×	17.10 ± 0.13	0.630 ± 0.009	0.453 ± 0.007
			✓	22.97 ± 0.32	0.801 ± 0.007	0.165 ± 0.006
	QVG	INT2	×	23.01 ± 0.27	0.826 ± 0.006	0.132 ± 0.004
			✓	**25.29** ± 0.32	**0.856** ± 0.005	**0.107** ± 0.004
SkyReels-V2	RTN	INT2	×	18.38 ± 0.16	0.670 ± 0.007	0.383 ± 0.007
			✓	18.39 ± 0.21	0.743 ± 0.008	0.269 ± 0.008
	QuaRot+RTN	INT2	×	19.20 ± 0.21	0.708 ± 0.009	0.319 ± 0.008
			✓	**20.42** ± 0.26	**0.784** ± 0.009	**0.202** ± 0.009
HY-WorldPlay	RTN	INT2	×	16.96 ± 0.66	0.573 ± 0.034	0.390 ± 0.030
			✓	17.15 ± 0.60	0.577 ± 0.030	0.311 ± 0.022
	QuaRot+RTN	INT2	×	17.16 ± 0.62	0.575 ± 0.034	0.376 ± 0.028
			✓	**18.27** ± 0.59	**0.616** ± 0.030	**0.273** ± 0.024

D.2 Statistical Significance and Prompt-Subset Robustness

All quantities in this subsection are computed from the per-prompt results behind Table 1; no additional generation runs are involved. All comparisons in Table 1 are paired: with and without correction on identical prompts, seeds, and, for HY-WorldPlay, actions. The first 30% of prompts from each VBench-Long dimension yield 286 prompts per model for MAGI-1 and SkyReels-V2. Two questions about the evaluation subset described in Section 5.1 can be answered from the per-prompt results: whether it is large enough to detect the effect of the correction reliably, and whether the gains are consistent across prompts. Table 6 addresses both. For each model, quantization scheme, and fidelity metric it reports the mean paired improvement, the paired t-test p-value (exact Wilcoxon signed-rank test, marked ^{W}, for HY-WorldPlay with n=10), and the per-sample win rate, i.e. the fraction of prompts on which the corrected output is closer to BF16. LPIPS is sign-flipped so that positive values always indicate improvement.

Table 6: Paired significance tests for the fidelity metrics in Table 1. Each cell gives the mean paired improvement, the p-value (paired t-test; ^{W}: exact Wilcoxon), and the fraction of samples improved. LPIPS is sign-flipped so that positive means improvement. n.s.: not significant at α=0.05 after Holm correction.

Model	Scheme	PSNR	SSIM	LPIPS
MAGI-1 (n=286)	RTN	+0.92, p=4×10^{-9}, 68%	+0.030, p=2×10^{-40}, 86%	+0.064, p=6×10^{-81}, 97%
	QuaRot+RTN	+5.87, p=6×10^{-67}, 93%	+0.171, p=9×10^{-102}, 99%	+0.288, p=6×10^{-140}, 100%
	QVG	+2.28, p=1×10^{-42}, 88%	+0.029, p=3×10^{-44}, 91%	+0.026, p=5×10^{-41}, 86%
SkyReels-V2 (n=286)	RTN	+0.004, p=0.98, 50% (n.s.)	+0.073, p=4×10^{-46}, 88%	+0.114, p=2×10^{-65}, 91%
	QuaRot+RTN	+1.22, p=3×10^{-15}, 70%	+0.077, p=4×10^{-57}, 95%	+0.117, p=2×10^{-68}, 92%
HY-WorldPlay (n=10)	RTN	+0.19, p=0.56^W, 60% (n.s.)	+0.004, p=0.70^W, 50% (n.s.)	+0.079, p=0.002^W, 100%
	QuaRot+RTN	+1.11, p=0.010^W, 90%	+0.041, p=0.002^W, 100%	+0.102, p=0.002^W, 100%

After Holm correction across all 21 comparisons, 18 remain significant: all 9 on MAGI-1, 5 of 6 on SkyReels-V2, and 4 of 6 on HY-WorldPlay. Across the significant comparisons, the correction improves 68–100% of individual samples, so the mean gains are not driven by a few outliers but are consistent across the evaluated prompts. Of the three inconclusive comparisons, SkyReels-V2 RTN PSNR reflects a negligible mean change (+0.004 dB) at full sample size, whereas the two HY-WorldPlay RTN cells (PSNR, SSIM) reflect the small n=10 set; LPIPS nonetheless improves on 10 of 10 videos there. These prompt-level results are consistent with the per-dimension VBench breakdown (Appendix D.3), which shows broad-based gains across the 16 VBench dimensions. The fixed selection rule prevents result-dependent prompt selection; a different 30% subset could still yield different effect sizes, and our claims therefore concern the paired effect of the correction on the evaluated prompts rather than absolute full-benchmark scores.

D.3 Per-Dimension VBench Results

Table 1 reports the aggregate VBench Score in the VBench-Long setting from VBench++ (Huang et al., 2024) on MAGI-1 and SkyReels-V2. For completeness, Tables 7 and 8 break this score down across all 16 VBench dimensions, grouped by VBench's *Quality* (visual fidelity) and *Semantic* (prompt fidelity) categories, and Table 9 reports the corresponding sub-scores together with the Total VBench Score that already appears in Table 1. All scores are reported with standard errors across prompts (± SE); within-prompt clips are averaged before computing the SE.

Table 7: Per-dimension VBench *Quality* results on MAGI-1 and SkyReels-V2 (subject consistency, background consistency, temporal flickering, motion smoothness, dynamic degree, aesthetic quality, imaging quality). Values are on the standard VBench 0–100 scale. ± denotes standard error across prompts. Best quantized result per model is **bolded**; ties are bolded jointly.

Model	Quant. scheme	Prec.	With corr.	Subj. Con.↑	BG Con.↑	Temp. Flick.↑	Mot. Smo.↑	Dyn. Deg.↑	Aes. Q.↑	Img. Q.↑
MAGI-1	—	BF16		98.24 ± 0.27	98.33 ± 0.15	99.64 ± 0.07	99.53 ± 0.04	18.46 ± 5.43	58.86 ± 2.09	58.63 ± 2.89
	RTN	INT2	×	97.71 ± 0.31	98.01 ± 0.07	99.57 ± 0.07	99.41 ± 0.05	15.38 ± 4.34	55.50 ± 2.12	54.50 ± 3.09
			✓	98.20 ± 0.23	**98.22** ± 0.08	**99.64** ± 0.07	**99.54** ± 0.03	15.38 ± 4.87	58.23 ± 2.04	57.84 ± 2.96
	QuaRot+RTN	INT2	×	94.26 ± 0.24	96.98 ± 0.09	99.10 ± 0.14	98.41 ± 0.23	**48.46** ± 5.57	40.71 ± 0.95	39.72 ± 1.78
			✓	98.01 ± 0.31	98.19 ± 0.12	99.61 ± 0.09	99.53 ± 0.04	16.92 ± 5.62	57.98 ± 1.98	58.16 ± 2.79
	QVG	INT2	×	97.93 ± 0.29	98.10 ± 0.11	99.51 ± 0.09	99.47 ± 0.04	17.69 ± 5.12	57.87 ± 2.08	57.78 ± 2.85
			✓	**98.24** ± 0.24	**98.22** ± 0.11	99.57 ± 0.09	99.52 ± 0.04	17.69 ± 5.47	**58.60** ± 2.05	**58.63** ± 2.85
SkyReels-V2	—	BF16		97.66 ± 0.30	97.46 ± 0.13	99.57 ± 0.09	99.15 ± 0.09	79.81 ± 6.05	53.87 ± 2.15	59.04 ± 2.18
	RTN	INT2	×	92.69 ± 0.37	95.75 ± 0.15	99.15 ± 0.05	98.04 ± 0.24	33.65 ± 6.79	41.23 ± 1.65	45.20 ± 2.52
			✓	96.98 ± 0.40	96.60 ± 0.18	**99.53** ± 0.06	98.94 ± 0.09	**85.58** ± 5.22	**56.06** ± 1.95	**63.38** ± 2.14
	QuaRot+RTN	INT2	×	94.13 ± 0.38	95.91 ± 0.18	99.34 ± 0.07	98.47 ± 0.18	51.92 ± 7.96	41.33 ± 1.73	48.43 ± 2.57
			✓	**97.56** ± 0.29	**97.05** ± 0.17	99.50 ± 0.10	**99.09** ± 0.09	81.73 ± 5.81	52.49 ± 2.06	58.22 ± 2.35

Table 8: Per-dimension VBench *Semantic* results on MAGI-1 and SkyReels-V2 (object class, multiple objects, human action, color, spatial relationship, scene, appearance style, temporal style, overall consistency). Values are on the standard VBench 0–100 scale. ± denotes standard error across prompts. Best quantized result per model is **bolded**; ties are bolded jointly.

Model	Quant. scheme	Prec.	With corr.	Obj. Cls.↑	Mult. Obj.↑	Hum. Act.↑	Color↑	Spat. Rel.↑	Scene↑	App. Sty.↑	Temp. Sty.↑	Overall Con.↑
MAGI-1	—	BF16		100.00 ± 0.00	57.40 ± 9.18	80.77 ± 7.88	96.00 ± 4.00	71.30 ± 7.61	18.70 ± 7.16	22.42 ± 0.69	22.38 ± 0.59	25.53 ± 1.14
	RTN	INT2	×	**100.00** ± 0.00	51.44 ± 9.28	76.92 ± 8.43	95.33 ± 4.02	69.10 ± 7.41	21.20 ± 7.14	**23.04** ± 0.66	20.83 ± 0.62	25.11 ± 1.20
			✓	**100.00** ± 0.00	55.10 ± 9.16	80.77 ± 7.88	95.55 ± 4.01	68.80 ± 8.19	**21.92** ± 7.72	22.53 ± 0.71	**22.48** ± 0.60	25.48 ± 1.15
	QuaRot+RTN	INT2	×	43.17 ± 3.59	18.85 ± 3.65	**88.46** ± 6.39	88.75 ± 4.14	25.41 ± 3.14	8.94 ± 3.30	23.00 ± 0.37	16.84 ± 0.85	22.67 ± 1.19
			✓	**100.00** ± 0.00	**55.48** ± 8.73	84.62 ± 7.22	**96.77** ± 4.00	**70.43** ± 8.03	20.82 ± 7.20	22.38 ± 0.71	22.37 ± 0.60	25.26 ± 1.22
	QVG	INT2	×	**100.00** ± 0.00	55.34 ± 9.28	84.62 ± 7.22	95.25 ± 4.04	67.81 ± 7.82	19.90 ± 7.19	22.66 ± 0.67	22.10 ± 0.59	25.43 ± 1.18
			✓	**100.00** ± 0.00	55.05 ± 9.15	84.62 ± 7.22	**96.77** ± 4.00	70.16 ± 7.57	21.25 ± 7.60	22.46 ± 0.71	22.22 ± 0.58	**25.51** ± 1.14
SkyReels-V2	—	BF16		75.30 ± 8.08	51.74 ± 8.60	76.92 ± 8.43	77.97 ± 7.40	67.41 ± 8.48	12.44 ± 6.31	18.86 ± 0.56	18.93 ± 0.98	21.01 ± 1.27
	RTN	INT2	×	46.69 ± 6.93	18.51 ± 3.77	65.38 ± 9.51	71.98 ± 5.48	34.87 ± 6.01	6.67 ± 3.24	**20.66** ± 0.42	20.10 ± 0.61	22.34 ± 0.95
			✓	**75.96** ± 7.30	50.78 ± 8.45	**80.77** ± 7.88	**77.71** ± 6.75	68.97 ± 7.86	**14.06** ± 6.61	18.82 ± 0.55	**20.34** ± 0.78	**22.73** ± 1.08
	QuaRot+RTN	INT2	×	59.68 ± 8.18	28.31 ± 5.15	61.54 ± 9.73	75.06 ± 6.10	45.87 ± 7.62	11.72 ± 5.09	19.86 ± 0.54	18.53 ± 0.84	20.54 ± 1.31
			✓	73.26 ± 8.11	**52.88** ± 8.40	76.92 ± 8.43	74.62 ± 7.45	**69.00** ± 7.63	13.46 ± 6.54	18.82 ± 0.56	19.35 ± 0.77	20.82 ± 1.27

Table 9: Aggregate VBench scores for MAGI-1 and SkyReels-V2: VBench's Quality and Semantic sub-scores and the total VBench Score (which already appears in Table 1). Values are on the standard VBench 0–100 scale. Best quantized result per model is **bolded**. ± denotes standard error across prompts, propagated to aggregate scores via linear error propagation through VBench's normalization and weighting.

Model	Quant. scheme	Prec.	With corr.	Quality↑	Semantic↑	Total↑
MAGI-1	—	BF16		80.10 ± 0.69	70.93 ± 1.97	78.27 ± 0.68
	RTN	INT2	×	78.46 ± 0.67	69.49 ± 2.00	76.67 ± 0.67
			✓	79.62 ± 0.67	70.83 ± 2.04	77.86 ± 0.67
	QuaRot+RTN	INT2	×	74.90 ± 0.55	51.62 ± 1.26	70.24 ± 0.50
			✓	79.69 ± 0.68	71.31 ± 1.94	78.02 ± 0.67
	QVG	INT2	×	79.57 ± 0.67	70.79 ± 1.96	77.81 ± 0.67
			✓	**79.95** ± 0.69	**71.35** ± 1.97	**78.23** ± 0.68
SkyReels-V2	—	BF16		83.60 ± 0.67	60.02 ± 2.27	78.89 ± 0.70
	RTN	INT2	×	73.97 ± 0.71	48.27 ± 1.74	68.83 ± 0.67
			✓	**84.62** ± 0.61	**61.71** ± 2.15	**80.04** ± 0.65
	QuaRot+RTN	INT2	×	76.48 ± 0.79	51.28 ± 2.06	71.44 ± 0.75
			✓	83.25 ± 0.66	59.91 ± 2.24	78.58 ± 0.69

D.4 KEY VERSUS VALUE QUANTIZATION

Our main experiments quantize both keys and values to evaluate practical KV-cache compression. To isolate the two, we additionally evaluate key-only and value-only quantization on MAGI-1 using the same VBench-Long subset and QuaRot+RTN configuration (Table 10). K-only and K+V quantization are nearly identical across all metrics. On VBench, V-only quantization remains statistically indistinguishable from BF16 given the reported standard errors. Thus, degradation in this configuration is dominated by the key path. Applying our correction under K-only quantization recovers $\sim 96\%$ of the VBench gap against the BF16 baseline; this matches the corrected K+V result within standard error. This supports the proposed mechanism: key noise perturbs logits before the convex exponential and softmax normalization, where it creates Jensen bias, while value noise enters only after attention weights are formed and is therefore outside the scope of our correction. We do not claim that all key-quantization error is caused by Jensen bias, but this ablation shows that keys dominate the degradation and that correcting their predicted bias recovers nearly all benchmark-level quality loss.

Table 10: Key-only and value-only INT2 quantization on MAGI-1 (QuaRot+RTN, g=32). $\pm$ denotes standard error across prompts. Degradation is dominated by the key path, and correcting the keys alone recovers nearly all of the quality loss.

Quantized states	Precision	With correction	PSNR↑	SSIM↑	LPIPS↓	VBench↑
—	BF16		–	–	–	78.27 ± 0.68
K+V	INT2	×	17.10 ± 0.13	0.630 ± 0.009	0.453 ± 0.007	70.24 ± 0.50
		✓	22.97 ± 0.32	0.801 ± 0.007	0.165 ± 0.006	78.02 ± 0.67
K-only	INT2	×	17.09 ± 0.13	0.632 ± 0.009	0.450 ± 0.007	70.41 ± 0.51
		✓	22.92 ± 0.33	0.800 ± 0.007	0.166 ± 0.006	77.98 ± 0.66
V-only	INT2	×	24.96 ± 0.41	0.838 ± 0.007	0.129 ± 0.006	78.44 ± 0.70

D.5 MEMORY–QUALITY TRADE-OFF FOR SSIM AND LPIPS

Figure 5 in the main text reports the memory–quality trade-off in terms of PSNR. For completeness, Figs. 6 and 7 report the same analysis for SSIM and LPIPS, confirming that the correction uniformly improves the trade-off across all three fidelity metrics.

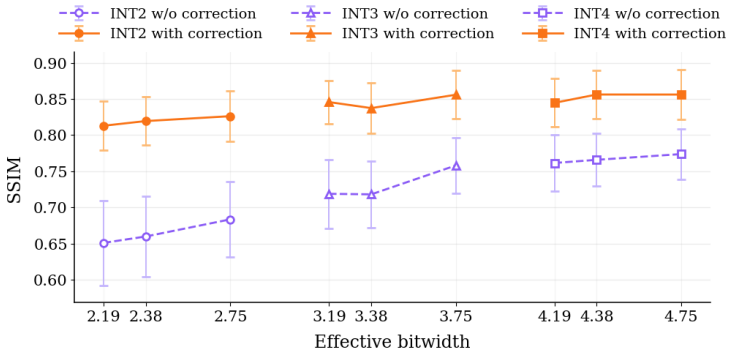


Figure 6: Trade-off between SSIM and effective bitwidth on MAGI-1. Same setting as Fig. 5.

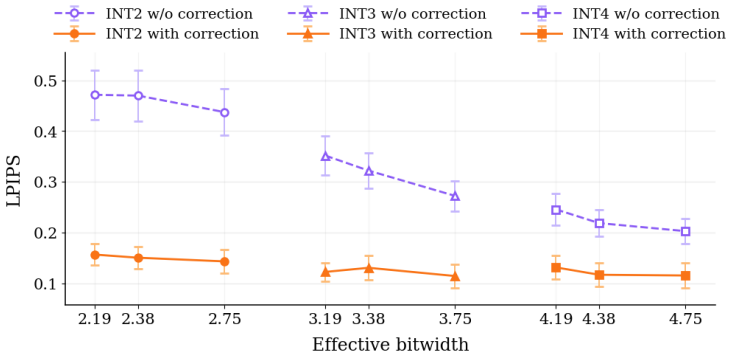


Figure 7: Trade-off between LPIPS and effective bitwidth on MAGI-1. Same setting as Fig. 5.

D.6 Qualitative Comparison on SkyReels-V2

Figure 1 in the main text shows the qualitative effect of INT2 KV-cache quantization and our correction on MAGI-1. Figure 8 reports the analogous comparison on SkyReels-V2 for two representative prompts from the VBench-Long suite.

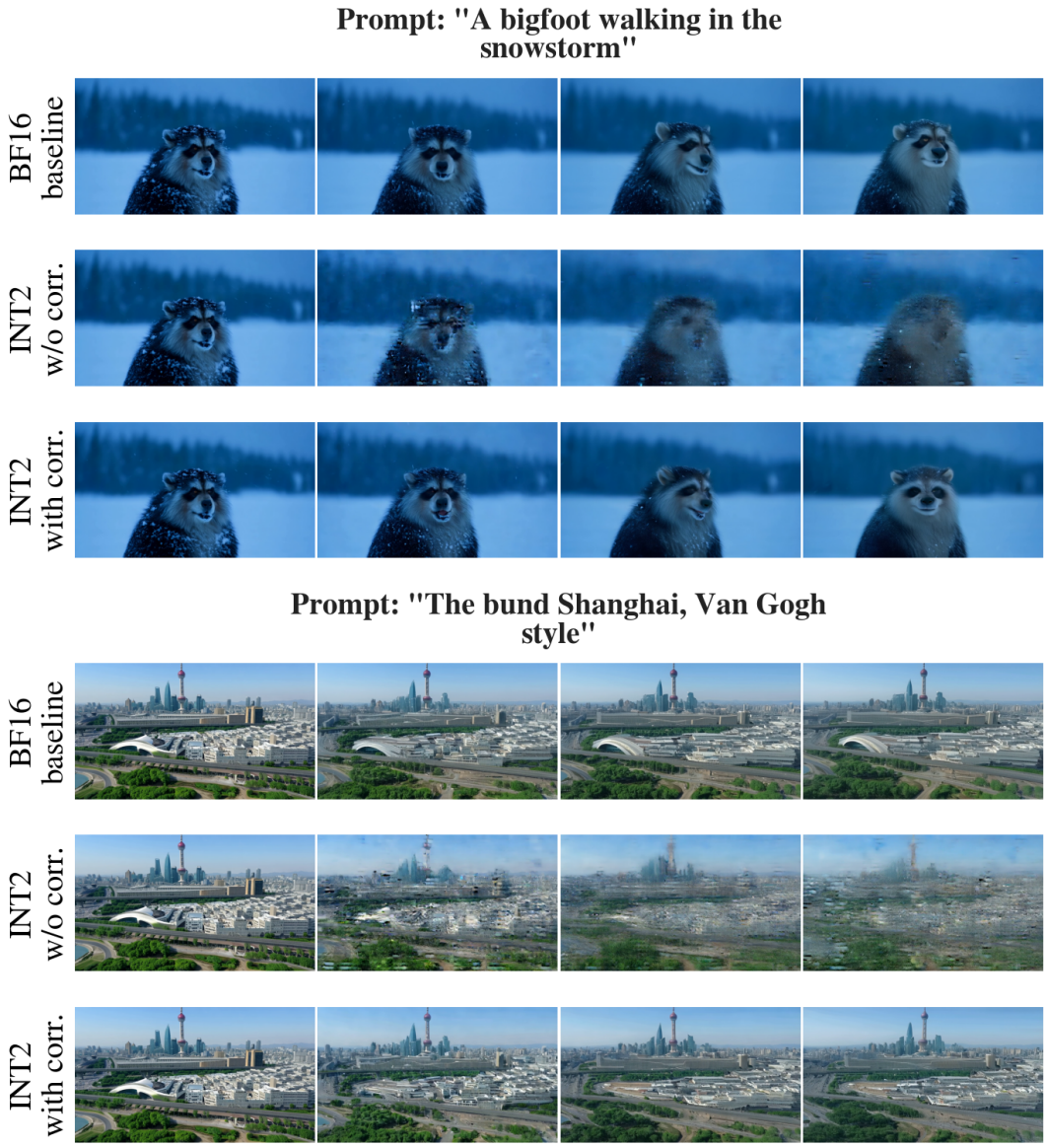


Figure 8: Qualitative comparison on SkyReels-V2. Columns show successive frames from the same video. Rows show BF16; INT2 asymmetric QuaRot+RTN quantization of cached keys and values; and the same setting with our correction. As on MAGI-1 (Fig. 1), INT2 introduces visible distortions, while our correction recovers much of the BF16-like visual quality and temporal consistency.

D.7 Qualitative Comparison on HY-WorldPlay

Figure 1 in the main text shows the qualitative effect of INT2 KV-cache quantization and our correction on MAGI-1. For completeness, Fig. 9 reports the analogous comparison on HY-WorldPlay for two representative image–prompt pairs from the original HY-WorldPlay repository.

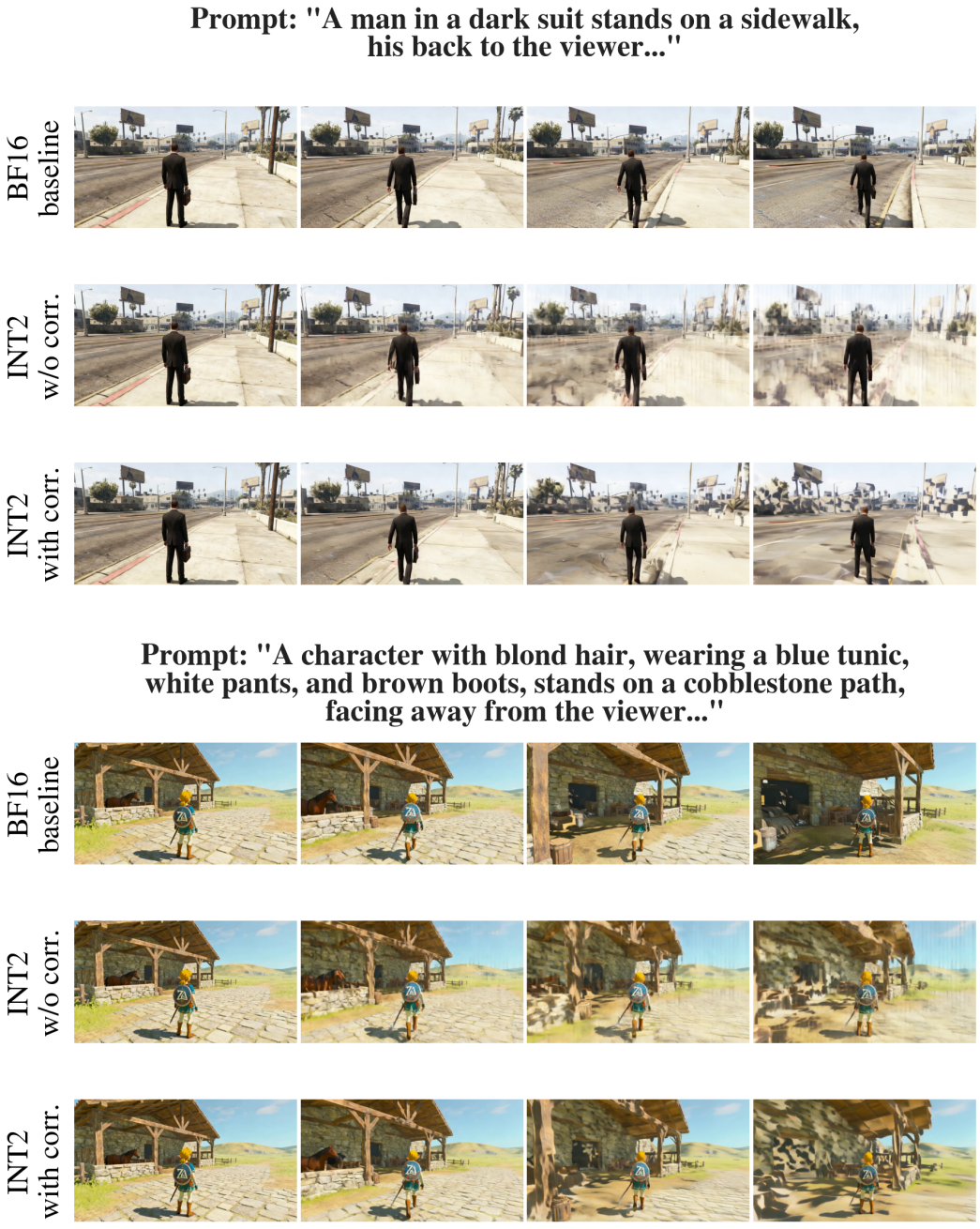


Figure 9: Qualitative comparison on HY-WorldPlay. Columns show successive frames from the same video. Rows show BF16; INT2 asymmetric QuaRot+RTN KV-cache quantization of keys and values; and the same quantized setting with our correction. As on MAGI-1 (Fig. 1), INT2 introduces visible distortions, while our correction recovers much of the BF16-like visual quality and temporal consistency.

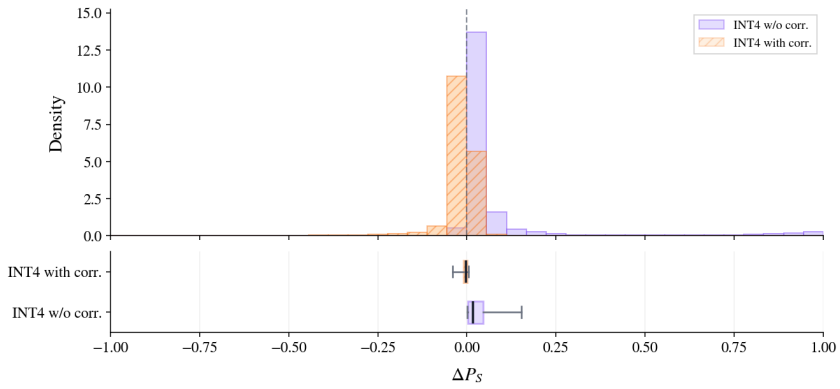
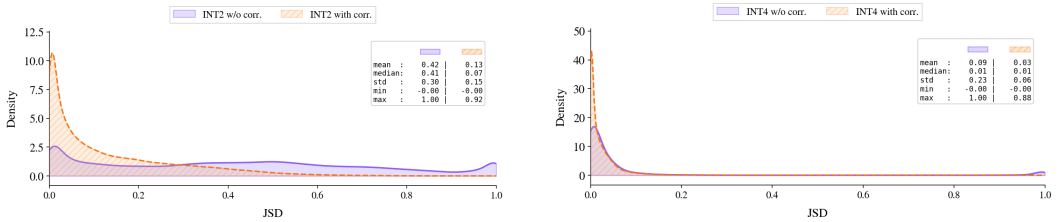


Figure 10: Cached attention-mass shift ΔP_S on MAGI-1 at INT4 with QuaRot+RTN KV-cache quantization. The same qualitative pattern as at INT2 (cf. Fig. 4) is visible, but the bias is much smaller. The correction centers the distribution near zero.



(a) INT2 quantization

(b) INT4 quantization

Figure 11: Distribution of Jensen–Shannon divergence between quantized (or corrected) and BF16 attention weights on MAGI-1 under QuaRot+RTN. At INT2 the correction substantially reduces the JSD; at INT4 the baseline JSD is already low and the improvement is modest.

E Additional Attention-Level Results

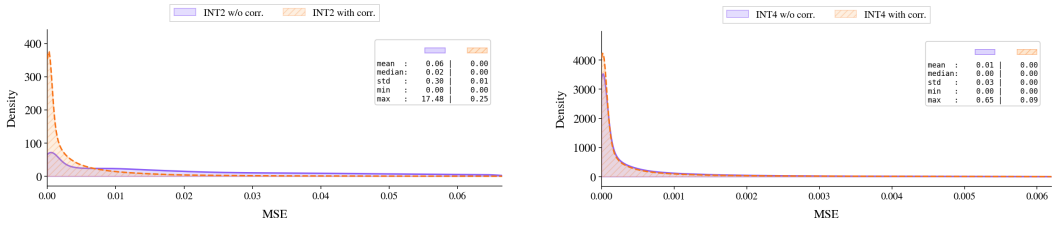
E.1 Attention Mass Shift at INT4

Figure 4 in the main text reports the cached attention-mass shift ΔP_S at INT2. For completeness, we report here the same analysis at INT4 on MAGI-1 with the same quantization scheme and our correction.

The INT4 results in Fig. 10 show the same qualitative pattern as at INT2: a right-skewed quantized distribution of ΔP_S that the correction centers near zero. However, the magnitude of the bias is much smaller. Because the uncorrected bias is already small at INT4 and generated videos are visually close to the BF16 baseline, the correction's benefit is correspondingly mild, which is why we focus the main paper on INT2.

E.2 Attention-Weight JSD

Figure 11 plots the distribution of Jensen–Shannon divergence (JSD) between the quantized (or corrected) and BF16 attention weights on MAGI-1 under QuaRot+RTN quantization, computed over all keys. At INT2, the correction consistently shifts the JSD distribution toward lower values, confirming that removing the partition-sum bias improves the overall attention distribution. At INT4 the JSD is already low without correction, and the correction provides only a modest further reduction, mirroring the smaller attention-mass shift observed in Appendix E.1.



(a) INT2 quantization

(b) INT4 quantization

Figure 12: Attention output MSE between quantized (or corrected) and BF16 computations on MAGI-1 under QuaRot+RTN. The correction reduces MSE at INT2, confirming that score-level improvements propagate to the attention output. At INT4 the effect is smaller.

E.3 Attention-Output MSE

Figure 12 reports the mean squared error (MSE) of the attention output $\mathrm{softmax}(S)\,V$ between the quantized (or corrected) and BF16 computations on MAGI-1 under QuaRot+RTN quantization. At INT2, the correction consistently reduces the attention output MSE, confirming that improvements at the score level propagate to the attention output. At INT4 the MSE follows the same trend as the JSD (Appendix E.2): already low without correction, with a modest further reduction after correction.

E.4 Joint Head-Level Analysis

The distributions reported at INT2 for the cached-mass shift $\Delta P_{\mathcal{S}}$ (Fig. 4), the attention-weight JSD (Fig. 11), and the attention-output MSE (Fig. 12) are marginal: each moves toward zero after correction, but none shows whether the three quantities are linked on the same attention heads. We therefore measured all three jointly on MAGI-1 under INT2, with 807,840 measurements per quantization scheme, where one measurement is one attention head at one prompt, denoising step, layer, and query chunk.

Table 11 sorts the QuaRot+RTN measurements by uncorrected absolute cached-mass shift $|\Delta P_{\mathcal{S}}|$ into ten equally sized groups. Both the JSD and the output MSE rise monotonically across all ten deciles, so the size of a head's cached-mass shift predicts how far its attention distribution and its output move from BF16 over the whole range.

Table 11: Median attention-weight JSD and attention-output MSE per decile of uncorrected absolute cached-mass shift $|\Delta P_{\mathcal{S}}|$ on MAGI-1 at INT2 (QuaRot+RTN, 807,840 head-level measurements).

Decile	Median $\lvert\Delta P_{\mathcal{S}}\rvert$	Median JSD	Median output MSE
1	0.0093	0.0046	0.000031
2	0.0722	0.0478	0.00125
3	0.1513	0.1210	0.00552
4	0.2336	0.2532	0.01326
5	0.3072	0.3283	0.01666
6	0.3775	0.3893	0.02179
7	0.4534	0.4574	0.02967
8	0.5415	0.5313	0.03839
9	0.6594	0.6604	0.05073
10	0.8757	0.9615	0.09146

Moreover, the correction's improvements co-occur on the same heads. For each measurement we compute the quantized-to-corrected improvement in cached-mass error, JSD, and output MSE. Table 12 reports the Spearman rank correlations between these per-measurement improvements: heads whose cached-mass error the correction reduces most are also the heads whose attention weights and outputs it improves most, under both quantization schemes.

These are correlational, head-level results and do not by themselves establish causality between attention stealing and end-to-end quality loss. Combined with the key-only and value-only ablation

Table 12: Spearman rank correlation between per-measurement improvements (quantized to corrected) in cached-mass error, attention-weight JSD, and attention-output MSE on MAGI-1 at INT2.

Pair of improvements	**QuaRot+RTN**	**RTN**
Cached-mass error vs. JSD	0.74	0.64
Cached-mass error vs. output MSE	0.55	0.46
JSD vs. output MSE	0.85	0.82

of Appendix D.4, however, they make key-induced attention stealing the most plausible primary mechanism behind the degradation.

F LLM CHUNKED-PREFILL EXPERIMENTS

Our main experiments focus on chunk-wise autoregressive video diffusion, where previously generated chunks are stored in a quantized KV cache and the current chunk remains in full precision. In this appendix, we evaluate whether our correction transfers to decoder-only language models under structurally analogous chunked prefill.

Following the notation of Section 4.1, each prompt contains a quantized cached prefix $\mathcal{S}$ and a full-precision current prefill chunk $\mathcal{R}$, where $|\mathcal{R}| \gg 1$. This setup preserves the key structural feature of chunk-wise video generation: a quantized cached block $\mathcal{S}$ competes inside the same softmax with a multi-token full-precision current block $\mathcal{R}$.

These experiments provide a cross-domain validation of the bias-correction mechanism derived in Section 4, rather than a comprehensive LLM inference benchmark.

F.1 EXPERIMENTAL SETUP

We evaluate three decoder-only LLMs: Llama-3.1-8B (Dubey et al., 2024; Meta, 2024), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023; Mistral AI, 2024), and Qwen2.5-32B-Instruct (Qwen et al., 2024; Qwen, 2024). We use English prompts from LongBench-Pro (Chen et al., 2026b). We define retained prompt-length bins, e.g., $[256, 512)$ and $[512, 1024)$, then deterministically truncate prompts to retained lengths sampled uniformly from the corresponding bin. Each evaluation job uses one fixed current-chunk size across the resulting mixed prompt lengths.

For each model and chunk size, we use the same INT2 KV-cache quantization as in the main paper. We apply our Taylor-approximated score correction to cached-key attention scores before softmax, as described in Section 4.

Completed runs cover current-chunk sizes from 128 to 8192 (4096 for Qwen2.5-32B-Instruct); larger attempted configurations exceeded accelerator memory even on 80 GB GPUs. This is due to the quadratic workspace of chunked-prefill attention, whose dense score tensor scales as $n_h|\mathcal{R}|(|\mathcal{S}|+|\mathcal{R}|)$, where n_h is the number of attention heads, $|\mathcal{S}|$ is the cached-prefix length, and $|\mathcal{R}|$ is the current-chunk length. To avoid artifacts from this missingness, all aggregate results are reported as paired comparisons: each difference is computed only within cells matched by model, current-chunk size, prompt-length bin, and evaluation examples.

Our primary metric is teacher-forced negative log-likelihood (NLL). For a set of evaluation examples $\mathcal{D}$, we aggregate at corpus level:

$$\mathrm{NLL} = \frac{\sum_{x \in \mathcal{D}} \sum_{t=1}^{T_x} -\log p_\theta(y_t \mid y_{<t}, x)}{\sum_{x \in \mathcal{D}} T_x}.$$

Here each example consists of a context x and T_x target tokens $y_1, \ldots, y_{T_x}$. We use NLL as the main metric because it aggregates token-level likelihoods directly and avoids the heavy-tailed behavior of averaging per-example perplexities.

F.2 LLM CHUNKED-PREFILL RESULTS

Figure 13 summarizes our findings for the LLM chunked-prefill study. Plain INT2 KV-cache quantization generally worsens teacher-forced NLL, while the Taylor correction improves over plain INT2 across all completed model and chunk-size settings. In most of these settings, the corrected condition is also below the BF16 NLL, although we interpret this conservatively as a chunked-prefill rebalancing effect rather than as evidence that the method generally improves over full precision.

We observe substantial degradation from INT2 KV-cache quantization, especially at large chunk sizes for the smaller Mistral-7B-Instruct-v0.3 and Llama-3.1-8B models. The larger Qwen2.5 model shows smaller plain-INT2 degradation, but the correction still consistently improves NLL. This suggests that the correction is useful both in severe degradation regimes and in milder regimes where plain INT2 remains relatively stable.

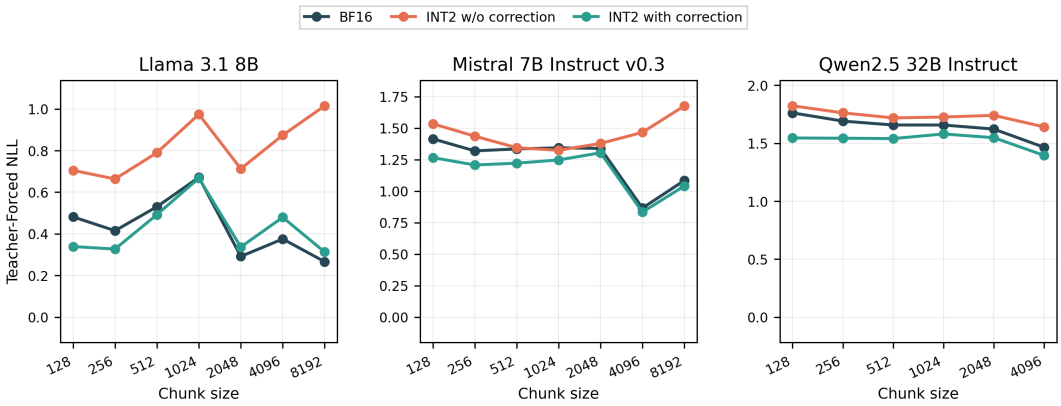


Figure 13: Teacher-forced NLL by chunk size in the LLM chunked-prefill setting. Each panel corresponds to one model, and curves show BF16, plain INT2 KV-cache quantization, and INT2 with Taylor correction. Plain INT2 generally increases NLL, while the Taylor correction consistently reduces NLL relative to plain INT2.

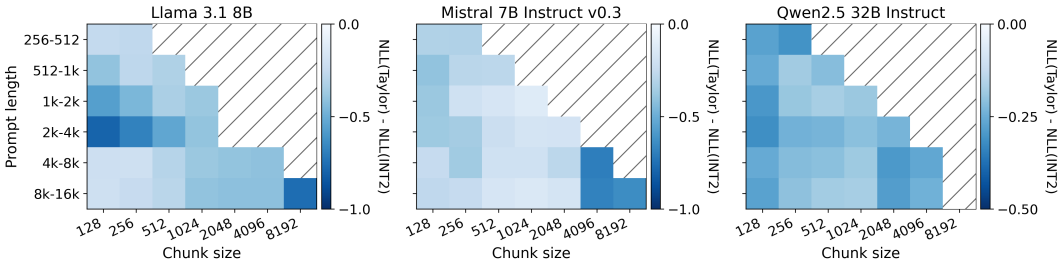


Figure 14: Prompt-length and chunk-size breakdown for LLM chunked-prefill experiments. The plotted value is $\mathrm{NLL}_{\mathrm{INT2+Taylor}} - \mathrm{NLL}_{\mathrm{INT2}}$, computed within matched model, chunk-size, prompt-bin, and evaluation-example cells. Negative values indicate that the Taylor correction reduces teacher-forced NLL relative to plain INT2 KV-cache quantization. Striped areas indicate no available matched data.

F.3 Prompt-Length and Chunk-Size Breakdown

To test whether the aggregate results are driven by a small subset of prompt lengths, we also analyze NLL by retained prompt-length bin. Figure 14 reports paired NLL differences grouped by prompt-length bin and current-chunk size.

F.4 Attention-Mass Diagnostic

The central mechanism studied in the main paper is that quantized cached keys receive inflated softmax mass because the exponential transforms zero-mean score noise into a positive partition-sum bias (Fig. 3; see also Fig. 2). Figure 15 visualizes the corresponding attention-weight shift in an LLM chunked-prefill setting.

For this diagnostic, we use Llama-3.2-1B as a lightweight model for attention visualization. This diagnostic model is separate from the three-model NLL benchmark above; it is used here because logging full attention weights across many layers, heads, prompts, and chunk sizes is memory intensive.

F.5 Discussion

The LLM chunked-prefill results provide additional evidence in a setting with a cached/current attention structure similar to that of the chunk-wise video diffusion experiments in Section 5. In the completed paired comparisons, plain INT2 KV-cache quantization generally worsens teacher-forced

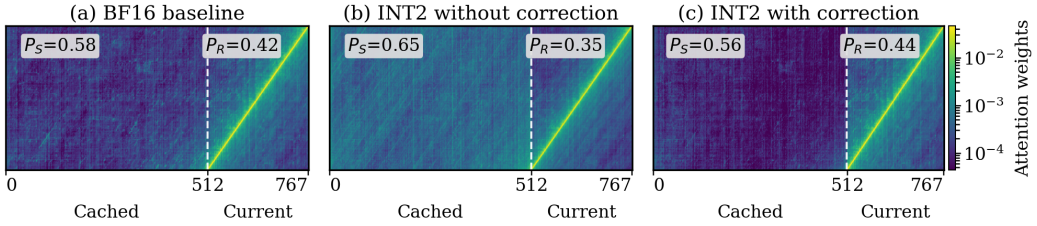


Figure 15: Attention weights for Llama-3.2-1B under INT2 KV-cache quantization. The visualized attention weights are averaged over layers, attention heads, and representative prompts with lengths in $[1024, 2048)$. They show a current chunk of 256 tokens with a 512-token cached prefix. The dashed vertical line separates cached-prefix tokens from current-chunk tokens. Panel **(b)** shows that, relative to the BF16 baseline in **(a)**, quantization increases attention weights in the cached block of tokens and decreases them in the current chunk. This effect is quantified by the attention masses $P_{\mathcal{S}}$ and $P_{\mathcal{R}}$ of the cached token block and current chunk. Panel **(c)** shows that our correction largely restores the original attention weights, with slight overcorrection.

NLL, while the Taylor correction reduces NLL relative to plain INT2. This trend is consistent with our derivation and video-model experiments, but we interpret the LLM results as a diagnostic extension rather than as a comprehensive LLM KV-cache quantization benchmark. We therefore emphasize paired teacher-forced NLL comparisons and leave optimized LLM kernels, broader task-level evaluation, and attention-mass diagnostics across more LLM models and chunk sizes to future work.

In most configurations, the corrected condition obtains lower NLL than the BF16 baseline. We treat this observation cautiously and do not interpret it as a general improvement over BF16. It may depend on the chunked-prefill setup, the teacher-forced NLL objective, or mild overcorrection from the Taylor approximation at aggressive bitwidths. Our main conclusion from these experiments is limited to the paired comparison between plain INT2 and INT2 with correction: the correction reduces the NLL degradation introduced by INT2 KV-cache quantization in the evaluated chunked-prefill settings.