Quantized Keys Steal Attention: Bias Correction for KV-Cache Compression in Video Diffusion

Tuna Tuncer [1 2] **Felix Becker** [2 †] **Thomas Pfeil** [2 †]

Abstract

Chunk-wise autoregressive video diffusion models rely on a KV cache of previously generated chunks to avoid redundant computation, but this cache quickly becomes a memory bottleneck as videos grow longer. This bottleneck is a central obstacle for efficient multimodal AI systems, where video and world models must maintain long temporal context for interactive simulation, planning, and embodied agents. Methods to 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 the cached partition sum: 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. Using a second-order Taylor approximation, the additional computational overhead is negligible, and no additional memory is needed 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.

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, where each chunk of frames is denoised independently and attends to previously generated chunks (Chen et al., 2024; 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. As a result, practical systems must limit the window size, which restricts the temporal information available to the model and can degrade long-range consistency (Lv et al., 2026). For multimodal agents that use generated video or world-model rollouts for prediction and planning, this becomes an agentic-memory bottleneck: the system must trade off visual context length against deployable memory and bandwidth.

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, we find that INT4 KV-cache quantization preserves reasonable quality, whereas reducing to INT2 leads to severely distorted frames (Figure 1, Figure 6, and Figure 7).

As a key source of this degradation, we identify that attention mass shifts heavily toward the cached tokens under aggressive quantization (Figure 3), an effect we consistently

[†]Felix Becker and Thomas Pfeil jointly supervised this work. [1]Technical University of Munich [2]Tensordyne. Correspondence to: Tuna Tuncer <tuna.tuncer@tum.de>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

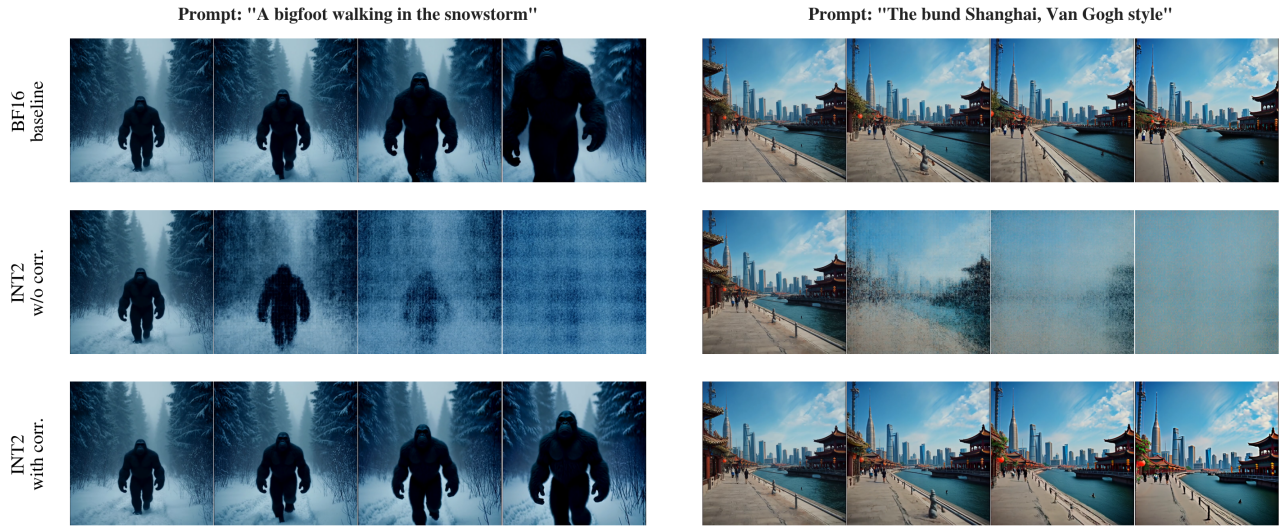


Figure 1. Qualitative comparison on MAGI-1 for two representative prompts. Columns show successive frames from the same generated video. From top to bottom: BF16 baseline; asymmetric INT2 (QuaRot+RTN) KV-cache quantization of both keys and values; same quantized setting with our correction. INT2 quantization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

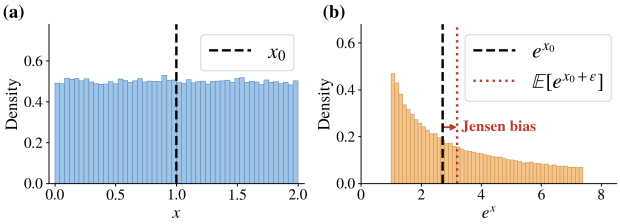


Figure 2. Illustration of the *Jensen bias* in exponentiated attention scores. Unquantized attention scores x_0 are subject to quantization noise $\varepsilon \sim \mathrm{Uniform}[-\Delta/2, \Delta/2]$. **(a)** In score space the noise is zero-mean, so the distribution of $x = x_0 + \varepsilon$ is centered at x_0. **(b)** After exponentiation the distribution becomes right-skewed: its mean $\mathbb{E}[e^{x_0+\varepsilon}]$ strictly exceeds $e^{x_0} = e^{\mathbb{E}[x_0+\varepsilon]}$. The gap $\mathbb{E}[e^{x_0+\varepsilon}] - e^{\mathbb{E}[x_0+\varepsilon]}$ is the Jensen bias, which our correction removes.

observe across layers, heads, and timesteps, and which correlates with poor video quality (Figure 1). The imbalance originates from how softmax transforms the noise introduced by quantizing the keys, as only keys—not values—participate in the exponentiation that defines attention weights. Integer quantization adds zero-mean noise to each cached token, so the resulting attention scores are also unbiased in expectation. However, once softmax exponentiates the scores, this symmetry breaks: because the exponential is convex, positive deviations are amplified more strongly than equal negative ones are suppressed. As a result, a symmetric noise distribution at the score level becomes right-skewed after exponentiation with its mean systematically above the exponential of the original, unquantized score (Figure 2).

We refer to this systematic, convexity-induced inflation as the *Jensen bias*; it is an instance of the Jensen gap studied in probability theory (Gao et al., 2020), here arising specifically in the softmax of quantized attention scores.

Our correction targets the Jensen bias directly. Because the bias is systematic rather than random, it can be estimated from quantities available at inference time. Before softmax, we subtract the correction factor from the scaled attention scores computed from cached keys. This restores the attention balance between cached and current tokens without retraining and without modifying the quantized KV cache values. In the representative attention example shown in Figure 3, the cached attention mass decreases from 0.95 under INT2 quantization to 0.46 after correction, close to the BF16 baseline of 0.59. The corresponding qualitative recovery in video quality is shown in Figure 1.

Our contributions are as follows:

- We identify the Jensen bias, a systematic inflation of attention mass on cached tokens caused by KV-cache quantization, and show that it is a major driver of quality degradation in autoregressive video diffusion under aggressive KV-cache compression.
- We derive a theoretically grounded, per-score correction and show that a simple Taylor approximation of this correction yields an effective and practical formula with negligible overhead.
- We demonstrate consistent benchmark improvements across three models, validating our proposed correction from attention weight level through to end-to-end video quality.

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) also uses random rotations, but studies KV-cache compression from an online vector-quantization perspective. It combines rotated-coordinate scalar quantization with a 1-bit Quantized Johnson–Lindenstrauss transform, yielding near-optimal distortion guarantees and unbiased inner-product estimates for low-bitwidth KV caches. 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 focus on key cache quantization. Our work is orthogonal to all the above references in that we do not improve the quantization representation itself, but instead analytically correct the systematic bias that any quantization scheme introduces in the softmax partition sum.

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 errors from quantized keys. In contrast, our correction is training-free and analytically derived. SageAttention (Zhang et al., 2025) 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-score correction shares.

Autoregressive video diffusion and efficient caching. Chunk-wise autoregressive video diffusion models generate videos by denoising successive chunks that attend to previously generated chunks through a KV cache (Chen et al., 2024; 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 error, we analytically correct the bias it introduces in softmax attention. We validate this complementarity empirically in Table 1, where composing the two methods on MAGI-1 yields the best overall results.

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

$$x_q = \mathrm{clamp}\big(\lfloor x/\Delta \rceil + z,\ 0,\ 2^B - 1\big), \tag{1}$$

and reconstructed as $\hat{x} = (x_q - z) \cdot \Delta$. The round-trip $x \mapsto x_q \mapsto \hat{x}$ introduces an additive error $\epsilon = \hat{x} - x$ that is bounded by $|\epsilon| \leq \Delta/2$. In practice, both Δ and z are chosen to cover the full $[\min, \max]$ range of the value being quantized.

Quantization granularity. The scale and zero-point can be shared at different granularities. In per-tensor quantization, one (Δ, z) pair is shared across an entire tensor. In per-token quantization, each token has its own (Δ_i, z_i). Group-wise per-token quantization further divides each token's d channels into groups of size g, with an independent $(\Delta_{i,j}, z_{i,j})$ per group j. The smaller the group of values sharing (Δ, z), the smaller the quantization error, but the larger the overall memory footprint.

Hadamard rotation. Key vectors in LLMs 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

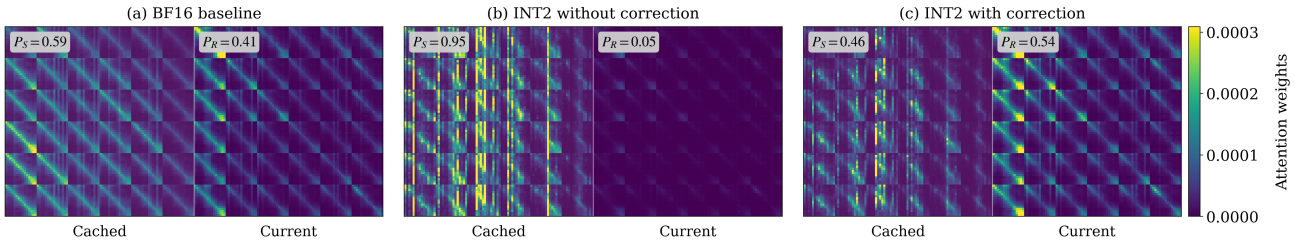


Figure 3. Attention weights for MAGI-1 for the prompt "a person" under INT2 KV-cache quantization. The visualization is taken from a representative layer, timestep, and attention head. Panel (b) shows that relative to the BF16 baseline in (a), quantization increases attention weights in the cached block of tokens and decreases them in the current chunk. This effect is quantified by the attention masses $P_{\mathcal{S}}$ and $P_{\mathcal{R}}$ of the cached token blocks and current chunks. (c) shows that our correction largely restores the original attention weights.

queries. The resulting more uniform distribution of values allows for lower errors in the subsequent quantization. Because H is orthogonal, the attention scores are preserved: $(Hq)^\top(Hk) = q^\top k$. For all ablation studies, we use such a Hadamard rotation before quantization, since this results in overall best quantized video quality.

Token structure. 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.

Attention at a denoising step. 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).

We now turn to the effect of quantization on this attention mechanism and derive a correction that compensates for the resulting bias in the softmax computation.

4. Method

We analyze the effect of KV-cache quantization on softmax attention and show that it introduces a systematic Jensen bias that inflates the contribution of cached keys. Based on this analysis, we derive a correction term that removes this bias in expectation, and present a practical approximation suitable for efficient implementation.

4.1. Score-Space Noise Under Quantization

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

$$s_i = \frac{q^\top k_i}{\sqrt{d}}, \qquad p_i = \frac{e^{s_i}}{\sum_{j=1}^{N} e^{s_j}}. \tag{2}$$

Recall from Section 3 that in autoregressive video generation, keys from previously generated chunks are quantized and stored in the KV cache, while keys of the current chunk have not yet been quantized. Let $\mathcal{S}$ denote the set of *cached*, quantized key indices and $\mathcal{R}$ the set of *current-chunk*, unquantized key indices, so that $\{1, \ldots, N\} = \mathcal{S} \cup \mathcal{R}$. We define the partition sums

$$Z_{\mathcal{S}} = \sum_{i \in \mathcal{S}} e^{s_i}, \qquad Z_{\mathcal{R}} = \sum_{i \in \mathcal{R}} e^{s_i}, \qquad Z = Z_{\mathcal{S}} + Z_{\mathcal{R}}. \tag{3}$$

We also define the total attention mass on the cached block,

$$P_{\mathcal{S}} = \sum_{i \in \mathcal{S}} p_i = \frac{Z_{\mathcal{S}}}{Z_{\mathcal{S}} + Z_{\mathcal{R}}}, \tag{4}$$

which measures how much attention mass is assigned to cached keys, and is what we ultimately care about when reasoning about attention stealing.

Quantization noise model. 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 rounding error. Following the standard statistical model for round-to-nearest uniform quantization (Widrow et al., 1996), we assume that the components are independent across channels $c \in \{1, \ldots, d\}$ and uniformly distributed:

$$\epsilon_{i,c} \sim \mathcal{U}\left(-\frac{\Delta_{i,c}}{2}, +\frac{\Delta_{i,c}}{2}\right). \tag{5}$$

Note that this noise model depends only on the round-to-nearest quantization operation itself, not on any prepro-

cessing applied to the keys before quantization (such as Hadamard rotations in QuaRot; see Section F).

The quantized attention score is then

$$\hat{s}_i = \frac{q^\top \hat{k}_i}{\sqrt{d}} = s_i + \delta_i, \qquad \delta_i = \frac{q^\top \epsilon_i}{\sqrt{d}}, \tag{6}$$

where δ_i is the score-space noise for key i. Under the uniform noise model, δ_i has zero mean and, by channel independence, its variance is

$$\sigma_i^2 = \mathrm{Var}(\delta_i) = \frac{1}{12\,d} \sum_{c=1}^{d} q_c^2\, \Delta_{i,c}^2. \tag{7}$$

For unquantized keys $i \in \mathcal{R}$, we have $\hat{s}_i = s_i$.

4.2. Quantization Bias in Softmax Attention

The score-space noise δ_i introduces a systematic bias in the softmax partition sum, which we now describe.

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:

$$\mathbb{E}\big[\hat{Z}_\mathcal{S}\big] = \sum_{i\in\mathcal{S}} e^{s_i} \cdot \mathbb{E}\big[e^{\delta_i}\big]. \tag{8}$$

For each term, Jensen's inequality applied to the convex function $\exp(\cdot)$ gives $\mathbb{E}[e^{\delta_i}] \geq e^{\mathbb{E}[\delta_i]} = 1$, so that $\mathbb{E}[\hat{Z}_\mathcal{S}] \geq Z_\mathcal{S}$ (see Figure 2 for an illustration of a single term). We call this systematic inflation of $\hat{Z}_\mathcal{S}$ the *Jensen bias*.

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

$$\Delta P_\mathcal{S} = \hat{P}_\mathcal{S} - P_\mathcal{S}, \qquad \hat{P}_\mathcal{S} = \frac{\hat{Z}_\mathcal{S}}{\hat{Z}_\mathcal{S} + Z_\mathcal{R}}. \tag{9}$$

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

Correction. We seek a deterministic per-score correction b_i, applied only to cached scores ($i \in \mathcal{S}$) and leaving current-chunk scores s_i ($i \in \mathcal{R}$) unchanged, that counteracts this bias. As shown above, 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:

$$e^{s_i - b_i} \cdot \mathbb{E}\big[e^{\delta_i}\big] \overset{!}{=} e^{s_i} \quad \Longrightarrow \quad \boxed{b_i = \log \mathbb{E}\big[e^{\delta_i}\big].} \tag{10}$$

Since every term is individually unbiased, the corrected cached partition sum is unbiased by linearity of expectation:

$$\mathbb{E}\big[\tilde{Z}_\mathcal{S}\big] = \sum_{i\in\mathcal{S}} e^{s_i - b_i} \cdot \mathbb{E}\big[e^{\delta_i}\big] = \sum_{i\in\mathcal{S}} e^{s_i} = Z_\mathcal{S}. \tag{11}$$

Note that $b_i \geq 0$ always (since $\mathbb{E}[e^{\delta_i}] \geq 1$ by Jensen's inequality), so the correction always *subtracts* from cached scores. Moreover, b_i is larger when the score-space noise is larger, i.e., when the relevant step sizes $\Delta_{i,c}$ are larger.

4.3. Computing the Correction

Since the noise components $\epsilon_{i,c}$ are independent across channels, the expectation $\mathbb{E}[e^{\delta_i}]$ factorizes across dimensions, leading to the exact correction term (for the full derivation, see Section A):

$$\boxed{b_i = \sum_{c=1}^{d} \log\left(\frac{\sinh\left(\dfrac{q_c\,\Delta_{i,c}}{2\sqrt{d}}\right)}{\dfrac{q_c\,\Delta_{i,c}}{2\sqrt{d}}}\right)} \tag{12}$$

Setting $\alpha_c = q_c \Delta_{i,c}/(2\sqrt{d})$ and using the second-order Taylor expansion $\log(\sinh(\alpha_c)/\alpha_c) \approx \alpha_c^2/6$ for small $|\alpha_c|$, this simplifies to:

$$\boxed{b_i \approx \frac{1}{24\,d} \sum_{c=1}^{d} q_c^2\, \Delta_{i,c}^2} \tag{13}$$

The Taylor approximation is simple, interpretable, and numerically stable. It shows that the bias scales with both the squared query magnitude and the squared quantization step size. We use this approximation in all experiments.

Connection to the noise variance. Comparing Equation (13) with Equation (7), the Taylor correction is exactly half the score-space noise variance: $b_i \approx \sigma_i^2/2$. This follows from the cumulant generating function (CGF). For any random variable X with cumulants $\kappa_1, \kappa_2, \kappa_3, \ldots$, the CGF satisfies

$$\log \mathbb{E}[e^X] = \kappa_1 + \frac{\kappa_2}{2} + \frac{\kappa_3}{6} + \cdots. \tag{14}$$

For zero-mean noise ($\kappa_1 = 0$), the leading term is $\kappa_2/2 = \sigma^2/2$, which depends only on the variance and not on the specific noise distribution. The exact closed-form correction in Equation (12) relies on the uniform noise model of integer quantization, but the second-order Taylor approximation requires only the score-space noise variance σ_i^2. This means that extending the correction to other quantization formats reduces to estimating σ_i^2 under the appropriate error model: for floating-point formats such as FP, MXFP, and NVFP, 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 $b_i \approx \sigma_i^2/2$.

Specialization to grouped per-token quantization. In our primary experimental setting, each token's d channels

are divided into $G = d/g$ groups of size g, and all channels within group j share the same step size $\Delta_{i,j}$. Writing $j(c)$ for the group containing channel c and $\|q_j\|^2 = \sum_{c \in \mathcal{G}_j} q_c^2$, substituting $\Delta_{i,c} = \Delta_{i,j(c)}$ into Equation (13) gives

$$b_i \approx \frac{1}{24\,d} \sum_{j=1}^{G} \Delta_{i,j}^2 \, \|q_j\|^2. \tag{15}$$

The same correction extends to QuaRot by replacing q with the rotated query Hq, so that $\|q_j\|^2$ becomes $\|(Hq)_j\|^2$ in Equation (13); we defer the derivation to Section F.

4.4. Overhead Analysis

We briefly analyze the storage and compute overhead. The interested reader is referred to the detailed investigation given in Section B.

Effective bitwidth. For group-wise quantization with group size g, the effective bitwidth is $B_{\text{eff}} = B + \frac{24}{g}$, accounting for per-group scale stored in FP8 and zero-point stored in BF16 metadata. Our correction introduces no additional storage, as it depends only on existing quantization parameters.

Computation at attention time. The Taylor correction adds an $O(QK \cdot d/g)$ term to attention computation, compared to the standard $O(QK \cdot d)$ cost of $QK^\top$. Thus, the additional work is reduced by a factor of g and is negligible in practice. In our FlexAttention-based implementation (Dong et al., 2024) on MAGI-1 with QuaRot+RTN and group size g=32, the correction adds approximately 5% end-to-end latency overhead relative to the quantized baseline.

5. Experiments

We evaluate the effectiveness of our proposed correction by measuring its impact both on attention behavior and on end-to-end video quality across multiple metrics and models.

5.1. Experimental Setup

Models. We evaluate our method on three autoregressive video diffusion models: MAGI-1 (Sand.ai et al., 2025) (4.5B), SkyReels-V2 (Chen et al., 2025) (1.3B), and HY-WorldPlay (Sun et al., 2025) (8B). 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 evaluate QuantVideoGen (QVG) (Xi et al., 2026) using its default configuration (S=1, B=64, K=256) to demonstrate that our correction composes with upstream video-aware cache compression. We apply the Taylor-approximated bias correction from Section 4.3 to all quantization schemes. Unquantized BF16 results serve as the reference outputs for fidelity metrics.

Table 1. Effect of the proposed correction on benchmark results for MAGI-1, SkyReels-V2, and HY-WorldPlay under INT2 KV-cache quantization. The correction consistently improves fidelity (PSNR) and perceptual quality (VBench), recovering much of the degradation introduced by quantization. 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. Full fidelity metrics (PSNR, SSIM, LPIPS) and per-dimension VBench scores with standard errors are reported in Tables 2 and 5.

Model	Scheme	Prec.	Corr.	PSNR↑	VBench↑
MAGI-1	—	BF16		–	78.27
	RTN	INT2	×	23.17	76.67
			✓	24.08	77.86
	QuaRot+RTN	INT2	×	17.10	70.24
			✓	22.97	78.02
	QVG	INT2	×	23.01	77.81
			✓	**25.29**	**78.23**
SkyReels-V2	—	BF16		–	78.89
	RTN	INT2	×	18.38	68.83
			✓	18.39	**80.04**
	QuaRot+RTN	INT2	×	19.20	71.44
			✓	**20.42**	78.58
HYWP	RTN	INT2	×	16.96	–
			✓	17.15	–
	QuaRot+RTN	INT2	×	17.16	–
			✓	**18.27**	–

Metrics. We report 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 videos. Table 1 reports PSNR and the aggregate VBench Score; a full breakdown across all metrics and all 16 VBench dimensions, together with VBench's Quality and Semantic sub-scores, is provided in Sections G and H.

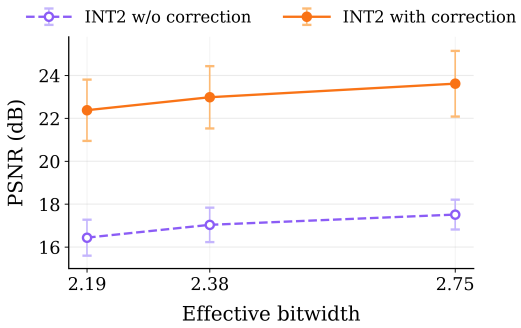


Figure 4. Trade-off between image quality, measured by PSNR, and memory footprint, measured as effective bitwidth per value, on MAGI-1 under INT2 KV-cache quantization. Bitwidths correspond to group sizes $g = \{128, 64, 32\}$.

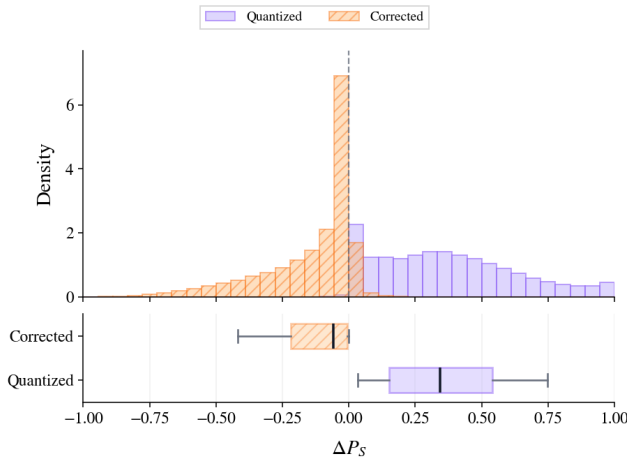


Figure 5. Shift in attention mass assigned to the cached block of tokens before (purple) and after (orange) our correction on MAGI-1 under INT2 KV-cache quantization. Positive values indicate that the quantized cached tokens steal attention from the current unquantized chunk. The median bias is large under INT2 quantization, and our correction significantly reduces this bias toward zero.

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. We do not evaluate on the full prompt set, as this is computationally prohibitive across all models and quantization configurations. 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.

5.2. Main Results

End-to-end quality. KV-cache quantization substantially degrades video quality across all three models on (Table 1), making most of the generated videos unusable in practice (see examples in Figure 1). Notably, the correction improves every reported metric in every evaluated configuration, without any model-specific tuning. On MAGI-1, composing our correction with QVG achieves the best results across all metrics, confirming that the two methods are complementary: QVG reduces the quantization error while our correction removes the residual Jensen bias. On MAGI-1 and SkyReels-V2, our correction closes the quality gap between INT2 KV-cache quantization and the BF16 baseline (Table 1). The consistent improvement across three distinct architectures empirically confirms the analysis in Section 4.2: the predicted bias is a meaningful driver of quality loss and the correction transfers across architectures without modification. The corrected videos of MAGI-1 and SkyReels-V2 reflect the perceptual quality of the BF16 reference while operating on a 2-bit KV cache. The per-dimension VBench breakdown in Section H confirms that this recovery on both models is broad-based across all 16 dimensions. HY-WorldPlay shows a smaller but consistent improvement, indicating that the same bias is at work but at a different scale.

Storage–quality trade-off. The effectiveness of our correction generalizes across different group sizes of INT2 quantization (Figure 4). It delivers a consistent PSNR gain across all group sizes we tested, so its benefit does not erode as one moves to coarser, more storage-efficient configurations. Hence, similar to INT2 quantization, our correction makes it easier to choose the trade-off between model quality and KV cache storage requirements.

5.3. Analysis

We validate our correction by showing that removing the Jensen bias improves metrics throughout the attention pipeline: from attention mass balance, to the full attention distribution (JSD; see Section L), to the attention output (MSE; see Section M), and ultimately to end-to-end video quality (PSNR, VBench). All results in this section are computed on MAGI-1 using a subset of prompts of each VBench dimension, averaged across all attention heads, layers, and denoising steps.

Attention mass shift. Attention stealing caused by the Jensen bias is illustrated in Figure 3. 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 5 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 Taylor approximation's behavior at aggressive bitwidths (Section A). Corresponding INT4 results are provided in Section K.

5.4. Cross-domain experiment: LLM partial 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 therefore run a small-scale diagnostic study on three decoder-only LLMs using LongBench-Pro English prompts (Chen et al., 2026b). We compare BF16, INT2 KV-cache quantization, and INT2 with our Taylor correction under teacher-forced negative log-likelihood (NLL), using paired model/chunk-size/prompt-length configurations.

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 N.

6. Discussion and Conclusion

We identified a systematic bias in softmax attention caused by KV-cache quantization. The Jensen bias inflates the attention mass on cached quantized keys, causing them to steal attention from tokens in the current chunk. We derived a correction term for each attention score that removes the Jensen bias in expectation, and presented a Taylor approximation that reduces computational complexity, rendering the computational overhead negligible. Experiments on MAGI-1, SkyReels-V2, and HY-WorldPlay demonstrate consistent fidelity improvements (PSNR, SSIM, LPIPS), with substantial VBench gains on MAGI-1 and SkyReels-V2, particularly at INT2 where quantization noise is most severe.

Limitations. Our experiments focus on chunked autoregressive video diffusion, where a multi-token current chunk competes with a quantized cached context. A similar Jensen bias can arise in standard LLM decoding, but it is less likely to be a dominant failure mode: many cached tokens compete with only one unquantized current token, so most attention mass is already on the cache. We therefore expect the correction to be most useful when the current unquantized block contains many tokens, as in chunked video generation or chunked LLM prefill.

We also note that our correction removes the Jensen bias only in expectation: when cached attention is relatively flat, the perturbations across many tokens average out so realizations stay close to expectation and the correction works as derived. However, when attention is strongly peaked on a few cached tokens, the law of large numbers no longer applies and noise on those few tokens can drive large gaps between realizations and expectation, limiting the gain of our correction. Independently, quantization schemes whose noise deviates from the approximately uniform, zero-mean model assumed here may also see reduced gains.

Future work. Because our correction acts purely in the space of attention scores, it is orthogonal to upstream KV-cache compression choices and composes with independent design axes such as the data type used for compression. Our derivation only requires a noise model for the quantization grid, so extending the correction to floating-point families such as FP, MXFP, and NVFP, whose non-uniform grids produce a different noise distribution, is a natural open direction.

A second substantive direction is extending the correction to LLM settings beyond standard single-token autoregressive decoding. Chunked prefill combined with KV-cache quantization (Gokhale et al., 2025) is a particularly promising setting, because each prefill chunk contains many current tokens, recovering a structure similar to the chunked video generation studied here. Our preliminary partial-prefill experiments support this direction, but a comprehensive LLM evaluation remains future work.

Impact Statement

This work improves the quality of low-bitwidth KV-cache quantization for autoregressive video diffusion, reducing the memory and energy cost of long-form video generation. Like other efficiency improvements for video generative models, it inherits the dual-use risks of the underlying technology, including non-consensual imagery and disinformation. Our method operates post-hoc on already-trained models and neither introduces new generative capabilities nor weakens existing safety mechanisms.

LLM Usage Statement

The authors used LLMs to assist with language polishing, including rephrasing, grammar correction, and improving readability, and occasionally for informal discussion of ideas during the course of the project. All technical contributions, including the identification of the Jensen bias, the derivation of the per-score correction, and the empirical evaluations, were developed and carried out by the authors, who take full responsibility for the final content of the manuscript.

References

Ashkboos, S., Mohtashami, A., Croci, M. L., Li, B., Jaggi, M., Alistarh, D., Hoefler, T., and Hensman, J. Quarot: Outlier-free 4-bit inference in rotated llms. *arXiv preprint arXiv:2404.00456*, 2024.

Chen, B., Monso, D. M., Du, Y., Simchowitz, M., Tedrake, R., and Sitzmann, V. Diffusion forcing: Next-token prediction meets full-sequence diffusion, 2024. URL https://arxiv.org/abs/2407.01392.

Chen, G., Lin, D., Yang, J., Lin, C., Zhu, J., Fan, M., Zhang, H., Chen, S., Chen, Z., Ma, C., Xiong, W., Wang, W., Pang, N., Kang, K., Xu, Z., Jin, Y., Liang, Y., Song, Y., Zhao, P., Xu, B., Qiu, D., Li, D., Fei, Z., Li, Y., and Zhou, Y. Skyreels-v2: Infinite-length film generative model, 2025. URL https://arxiv.org/abs/2504.13074.

Chen, S., Wei, C., Sun, S., Nie, P., Zhou, K., Zhang, G., Yang, M.-H., and Chen, W. Context forcing: Consistent autoregressive video generation with long context, 2026a. URL https://arxiv.org/abs/2602.06028.

Chen, Z., Wu, X., Jia, J., Gao, C., Fu, Q., Zhang, D., and Hu, S. Longbench pro: A more realistic and comprehensive bilingual long-context evaluation benchmark, 2026b. URL https://arxiv.org/abs/2601.02872.

Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. Llm.int8(): 8-bit matrix multiplication for transformers at scale, 2022. URL https://arxiv.org/abs/2208.07339.

Dong, J., Feng, B., Guessous, D., Liang, Y., and He, H. Flex attention: A programming model for generating optimized attention kernels, 2024. URL https://arxiv.org/abs/2412.05496.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Gao, X., Sitharam, M., and Roitberg, A. E. 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.

Gokhale, S., Das, D., Patwari, R., Sirasao, A., and Delaye, E. Kv pareto: Systems-level optimization of kv cache and model compression for long context inference, 2025. URL https://arxiv.org/abs/2512.01953.

Hooper, C., Kim, S., Mohammadzadeh, H., Mahoney, M. W., Shao, Y. S., Keutzer, K., and Gholami, A. Kvquant: Towards 10 million context length llm inference with kv cache quantization. *arXiv preprint arXiv:2401.18079*, 2024.

Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., Wang, Y., Chen, X., Wang, L., Lin, D., Qiao, Y., and Liu, Z. Vbench: Comprehensive benchmark suite for video generative models, 2023. URL https://arxiv.org/abs/2311.17982.

Huang, Z., Zhang, F., Xu, X., He, Y., Yu, J., Dong, Z., Ma, Q., Chanpaisit, N., Si, C., Jiang, Y., Wang, Y., Chen, X., Chen, Y.-C., Wang, L., Lin, D., Qiao, Y., and Liu, Z. Vbench++: Comprehensive and versatile benchmark suite for video generative models, 2024. URL https://arxiv.org/abs/2411.13503.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Le Scao, T., Lavril, T., Wang, T., Lacroix, T., and El Sayed, W. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. URL https://arxiv.org/abs/2310.06825.

Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., Wu, K., Lin, Q., Yuan, J., Long, Y., Wang, A., Wang, A., Li, C., Huang, D., Yang, F., Tan, H., Wang, H., Song, J., Bai, J., Wu, J., Xue, J., Wang, J., Wang, K., Liu, M., Li, P., Li, S., Wang, W., Yu, W., Deng, X., Li, Y., Chen, Y., Cui, Y., Peng, Y., Yu, Z., He, Z., Xu, Z., Zhou, Z., Xu, Z., Tao, Y., Lu, Q., Liu, S., Zhou, D., Wang, H., Yang, Y., Wang, D., Liu, Y., Jiang, J., and Zhong, C. Hunyuanvideo: A systematic framework for large video generative models, 2025. URL https://arxiv.org/abs/2412.03603.

Kwon, W. et al. Efficient memory management for large language model serving with pagedattention. *arXiv preprint arXiv:2309.06180*, 2023.

Liu, Z., Yuan, J., Jin, H., Zhong, S., Xu, Z., Braverman, V., Chen, B., and Hu, X. 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.

Lv, C., Shi, Y., Huang, Y., Gong, R., Ren, S., and Wang, W. Light forcing: Accelerating autoregressive video diffusion via sparse attention, 2026. URL https://arxiv.org/abs/2602.04789.

Ma, Y., Zheng, X., Xu, J., Xu, X., Ling, F., Zheng, X., Kuang, H., Li, H., Wang, X., Xiao, X., Chao, F., and Ji, R. 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.

Pandey, N. P., Fournarakis, M., Patel, C., and Nagel, M. 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, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Samuel, D., Tzachor, I., Levy, M., Green, M., Chechik, G., and Ben-Ari, R. Fast autoregressive video diffusion and world models with temporal cache compression and sparse attention, 2026. URL https://arxiv.org/abs/2602.01801.

Sand.ai, Teng, H., Jia, H., Sun, L., Li, L., Li, M., Tang, M., Han, S., Zhang, T., Zhang, W. Q., Luo, W., Kang, X., Sun, Y., Cao, Y., Huang, Y., Lin, Y., Fang, Y., Tao, Z., Zhang, Z., Wang, Z., Liu, Z., Shi, D., Su, G., Sun, H., Pan, H., Wang, J., Sheng, J., Cui, M., Hu, M., Yan, M., Yin, S., Zhang, S., Liu, T., Yin, X., Yang, X., Song, X., Hu, X., Zhang, Y., and Li, Y. Magi-1: Autoregressive video generation at scale, 2025. URL https://arxiv.org/abs/2505.13211.

Saxena, U. and Roy, K. Kvlinc : Kv cache quantization with hadamard rotation and linear correction, 2025. URL https://arxiv.org/abs/2510.05373.

Sun, W., Zhang, H., Wang, H., Wu, J., Wang, Z., Wang, Z., Wang, Y., Zhang, J., Wang, T., and Guo, C. Worldplay: Towards long-term geometric consistency for real-time interactive world modeling, 2025. URL https://arxiv.org/abs/2512.14614.

Tao, Q., Yu, W., and Zhou, J. Asymkv: Enabling 1-bit quantization of kv cache with layer-wise asymmetric quantization configurations, 2024. URL https://arxiv.org/abs/2410.13212.

Team Wan, Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.-W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Feng, M., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z.-F., and Liu, Z. Wan: Open and advanced large-scale video generative models, 2025. URL https://arxiv.org/abs/2503.20314.

Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13 (4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

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

Xi, H., Yang, S., Zhao, Y., Li, M., Cai, H., Li, X., Lin, Y., Zhang, Z., Zhang, J., Li, X., Xu, Z., Wu, J., Xu, C., Stoica, I., Han, S., and Keutzer, K. Quant videogen: Autoregressive long video generation via 2-bit kv-cache quantization, 2026. URL https://arxiv.org/abs/2602.02958.

Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., Yin, D., Zhang, Y., Wang, W., Cheng, Y., Xu, B., Gu, X., Dong, Y., and Tang, J. Cogvideox: Text-to-video diffusion models with an expert transformer, 2025. URL https://arxiv.org/abs/2408.06072.

Yao, Y., Tian, F., Chen, J., Lin, H., Dai, G., Liu, Y., and Wang, J. Timestep-aware correction for quantized diffusion models, 2024. URL https://arxiv.org/abs/2407.03917.

Yin, T., Zhang, Q., Zhang, R., Freeman, W. T., Durand, F., Shechtman, E., and Huang, X. From slow bidirectional to fast autoregressive video diffusion models, 2025. URL https://arxiv.org/abs/2412.07772.

Zandieh, A., Daliri, M., Hadian, M., and Mirrokni, V. Turboquant: Online vector quantization with near-optimal distortion rate, 2025. URL https://arxiv.org/abs/2504.19874.

Zhang, J., Huang, H., Zhang, P., Wei, J., Zhu, J., and Chen, J. Sageattention2: Efficient attention with thorough outlier smoothing and per-thread int4 quantization. In *International Conference on Machine Learning (ICML)*, 2025.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric, 2018. URL https://arxiv.org/abs/1801.03924.

A. Exact Correction: Full Derivation

We derive the exact formula for $b_i = \log \mathbb{E}[e^{\delta_i} \mid \{\Delta_{i,c}\}]$ under the uniform quantization noise model of Section 4.1.

Recall that $\delta_i = \sum_{c=1}^{d} q_c\, \epsilon_{i,c}/\sqrt{d}$, where the $\epsilon_{i,c}$ are independent with $\epsilon_{i,c} \sim \mathcal{U}(-\Delta_{i,c}/2, +\Delta_{i,c}/2)$. By independence across channels, the moment generating function factorizes:

$$\mathbb{E}\big[e^{\delta_i}\big] = \prod_{c=1}^{d} \mathbb{E}\left[\exp\left(\frac{q_c\, \epsilon_{i,c}}{\sqrt{d}}\right)\right]. \tag{16}$$

For each channel c, we evaluate the scalar MGF. Let $t_c = q_c/\sqrt{d}$ for brevity. Since $\epsilon_{i,c} \sim \mathcal{U}(-\Delta_{i,c}/2,\ +\Delta_{i,c}/2)$:

$$\begin{aligned}\mathbb{E}\big[e^{t_c\, \epsilon_{i,c}}\big] &= \frac{1}{\Delta_{i,c}} \int_{-\Delta_{i,c}/2}^{+\Delta_{i,c}/2} e^{t_c\, u}\, du \\ &= \frac{\sinh(t_c\, \Delta_{i,c}/2)}{t_c\, \Delta_{i,c}/2}.\end{aligned} \tag{17}$$

Taking the product over all channels and then the logarithm yields the exact correction:

$$b_i = \sum_{c=1}^{d} \log\left(\frac{\sinh\left(\dfrac{q_c\, \Delta_{i,c}}{2\sqrt{d}}\right)}{\dfrac{q_c\, \Delta_{i,c}}{2\sqrt{d}}}\right). \tag{18}$$

A naive implementation of this formula is numerically unstable ($\sinh$ overflows for large arguments) and computationally expensive ($O(d)$ operations per score entry, matching the attention score computation itself). We therefore seek a cheaper approximation.

Taylor approximation. Let $\alpha_c = q_c\, \Delta_{i,c}/(2\sqrt{d})$. Using $\log(\sinh(\alpha)/\alpha) = \alpha^2/6 + O(\alpha^4)$, and summing over channels:

$$b_i \approx \sum_{c=1}^{d} \frac{\alpha_c^2}{6} = \frac{1}{24\, d} \sum_{c=1}^{d} q_c^2\, \Delta_{i,c}^2. \tag{19}$$

Under group-wise per-token quantization, where each token's d channels are divided into $G = d/g$ groups sharing a common step size $\Delta_{i,j}$, this simplifies to $b_i \approx \frac{1}{24d} \sum_{j=1}^{G} \Delta_{i,j}^2\, \|q_j\|^2$ as derived in Section 4.3.

At aggressive bitwidths (e.g., INT2), the approximation may overcorrect, but we find empirically that this generally does not harm end-to-end video quality (see Section 5).

B. Detailed 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.

Under group-wise quantization with $G = d/g$ groups:

Per-query: Compute $\|q_j\|^2 = \sum_{c \in G_j} q_c^2$ 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 key.

Per score entry: Compute an inner product between 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:

$$O(Q \cdot d + K \cdot G + Q \cdot K \cdot G).$$

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

$$B_{\text{eff}} = \frac{d \cdot B + 24 \cdot G}{d} = B + \frac{24}{g}. \tag{20}$$

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. Implementation Note

In our implementation, the correction subtracts a per-score value b_i from cached scores before softmax. Materializing this correction for every score entry would require a dense tensor with the same shape as the full score matrix, which is unnecessary for long contexts. Instead, we apply the bias on the fly through a `score_mod` function in PyTorch's FlexAttention (Dong et al., 2024), which lets the fused attention kernel incorporate the correction without materializing the full correction tensor.

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

D. Pseudocode for Taylor-Corrected Attention

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

Require: Query matrix $Q \in \mathbb{R}^{M \times d}$; cached quantized keys $K_{\mathcal{S}}^{q}$ with per-group step sizes $\{\Delta_{i,j}\}$; cached values $V_{\mathcal{S}}$; current-chunk keys $K_{\mathcal{R}}$; current-chunk values $V_{\mathcal{R}}$; group size g, number of groups $G = d/g$

Ensure: Attention output $O \in \mathbb{R}^{M \times d_v}$

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

E. Per-Channel Quantization Correction

When quantization is performed per-channel (or per-group along the channel axis), the step size Δ_c depends on channel c but is shared across all tokens. The noise model becomes $\epsilon_{i,c} \sim \mathcal{U}(-\Delta_c/2,\ +\Delta_c/2)$, independent across channels and identically distributed across tokens for each fixed channel.

Since $\{\Delta_c\}$ do not depend on i, the distribution of $\delta_i = \sum_c q_c\, \epsilon_{i,c}/\sqrt{d}$ is the same for all cached keys $i \in \mathcal{S}$. The correction reduces to a single scalar shared by all tokens:

$$b = \sum_{c=1}^{d} \log \left(\frac{\sinh\left(\frac{q_c\, \Delta_c}{2\sqrt{d}}\right)}{\frac{q_c\, \Delta_c}{2\sqrt{d}}} \right), \tag{21}$$

with the Taylor approximation

$$b \approx \frac{1}{24\, d} \sum_{c=1}^{d} q_c^2\, \Delta_c^2. \tag{22}$$

Per-channel correction. Since b is the same for all $i \in \mathcal{S}$, the corrected scores within the cached chunk are $\tilde{s}_i = \hat{s}_i - b$ for all $i \in \mathcal{S}$. Subtracting b from all cached scores reduces $Z_{\mathcal{S}}$ relative to $Z_{\mathcal{R}}$, restoring the inter-chunk attention balance.

Under per-token quantization, the correction b_i varies across tokens, allowing it to differentially adjust each token's contribution. In our experiments, per-token quantization with the token-dependent correction consistently outperforms per-channel quantization with a shared correction.

F. Extension to QuaRot

Our derivation so far has been in the unrotated space. We now extend the correction to QuaRot (see Section 3).

With the Hadamard matrix H applied to both keys and queries, the quantized score becomes

$$\hat{s}_i = \frac{(Hq)^\top (Hk_i + \epsilon_i)}{\sqrt{d}} = s_i + \delta_i^{(H)}, \tag{23}$$

where $\delta_i^{(H)} = (Hq)^\top \epsilon_i/\sqrt{d}$. Our correction applies identically with q replaced by Hq: $b_i^{(H)} = \log \mathbb{E}[e^{\delta_i^{(H)}}]$.

Taylor approximation under rotation. The Taylor approximation replaces $\|q_j\|^2$ with $\|(Hq)_j\|^2$ (the per-group squared norms of the rotated query):

$$b_i^{(H)} \approx \frac{1}{24\, d} \sum_{j=1}^{G} \Delta_{i,j}^2\, \|(Hq)_j\|^2. \tag{24}$$

Note that while $\|Hq\|^2 = \|q\|^2$ by orthogonality, the per-group norms $\|(Hq)_j\|^2$ generally differ from $\|q_j\|^2$ because Hadamard rotation mixes channels across groups.

G. Fidelity Metric Standard Errors

Table 1 reports PSNR averaged across prompts. Table 2 reports the full set of fidelity metrics (PSNR, SSIM, LPIPS) with standard errors computed across prompts (the independent sampling unit), using the evaluation data described in Section 5.1.

H. Per-Dimension VBench Results

Table 1 reports the aggregate VBench Score on MAGI-1 and SkyReels-V2. For completeness, Tables 3 and 4 break this score down across all 16 VBench dimensions, grouped by VBench's *Quality* (visual fidelity) and *Semantic* (prompt fidelity) categories, and Table 5 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 ($\pm$ SE); within-prompt clips are averaged before computing the SE.

I. 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 6 reports the analogous comparison on SkyReels-V2 for two representative prompts from the VBench-Long suite.

Table 2. Fidelity metrics with standard errors for all configurations in Table 1. PSNR, SSIM, and LPIPS are computed relative to the BF16 reference; ± 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

J. 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, Figure 7 reports the analogous comparison on HY-WorldPlay for two representative image–prompt pairs from the original HY-WorldPlay repository.

K. Attention Mass Shift

Figure 5 in the main text reports the cached attention mass shift $\Delta P_{\mathcal{S}}$ 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 Figure 8 show the same qualitative pattern as at INT2: a right-skewed quantized distribution of $\Delta P_{\mathcal{S}}$ 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.

L. Attention JSD Distributions

We plot the distribution of Jensen-Shannon divergence (JSD) between the quantized (or corrected) and BF16 attention weights, computed over all keys. 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 probability-mass bias observed in Section K.

M. Attention Output MSE

We measure the mean squared error (MSE) of the attention output $\mathrm{softmax}(S)\,V$ between the quantized (or corrected) and BF16 computations. 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 (Section L): already low without correction, with a modest further reduction after correction.

Table 3. 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**.

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 4. 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

N. LLM Partial-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 partial prefill.

Following the notation of Section 4.2, each prompt contains a quantized cached prefix $\mathcal{S}$ and a full-precision current prefill chunk $\mathcal{R}$, with lengths $|\mathcal{S}| = A$ and $|\mathcal{R}| = B$, where $B \gg 1$. 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.

N.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., $[256, 512)$, $[512, 1024)$, etc., then deterministically truncate prompts to retained lengths sampled uniformly from the corresponding bin. Each evaluation job

Table 5. 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

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 4.

Completed runs cover current-chunk sizes from 128 to 8192; larger attempted configurations exceeded accelerator memory even on 80 GB GPUs. This is due to the quadratic workspace of partial-prefill attention, whose dense score tensor scales as $HB(A + B)$, where H is the number of attention heads, $A = |\mathcal{S}|$ is the cached-prefix length, and $B = |\mathcal{R}|$ 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:

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

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.

N.2. LLM partial prefill results

Figure 11 summarizes our findings for the LLM ablation study. Plain INT2 KV-cache quantization consistently worsens teacher-forced NLL, while the Taylor correction improves over plain INT2 across the completed model and chunk-size settings. The corrected condition is sometimes below the BF16 NLL, although we interpret this conservatively as a partial-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-3.2-1B 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.

N.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 12 reports paired NLL differences grouped by prompt-length bin and current-chunk size.

N.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. 2; see also Fig. 3). Figure 13 visualizes the corresponding attention-weight shift in an LLM partial-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.

N.5. Discussion

The LLM partial-prefill results provide additional indication in a cached/current attention structure setting similar to the main experiments on chunked auto-regressive video diffusion in 5. In the completed paired comparisons, plain INT2 KV-cache quantization generally worsens teacher-forced 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 some 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 partial-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 partial-prefill settings.

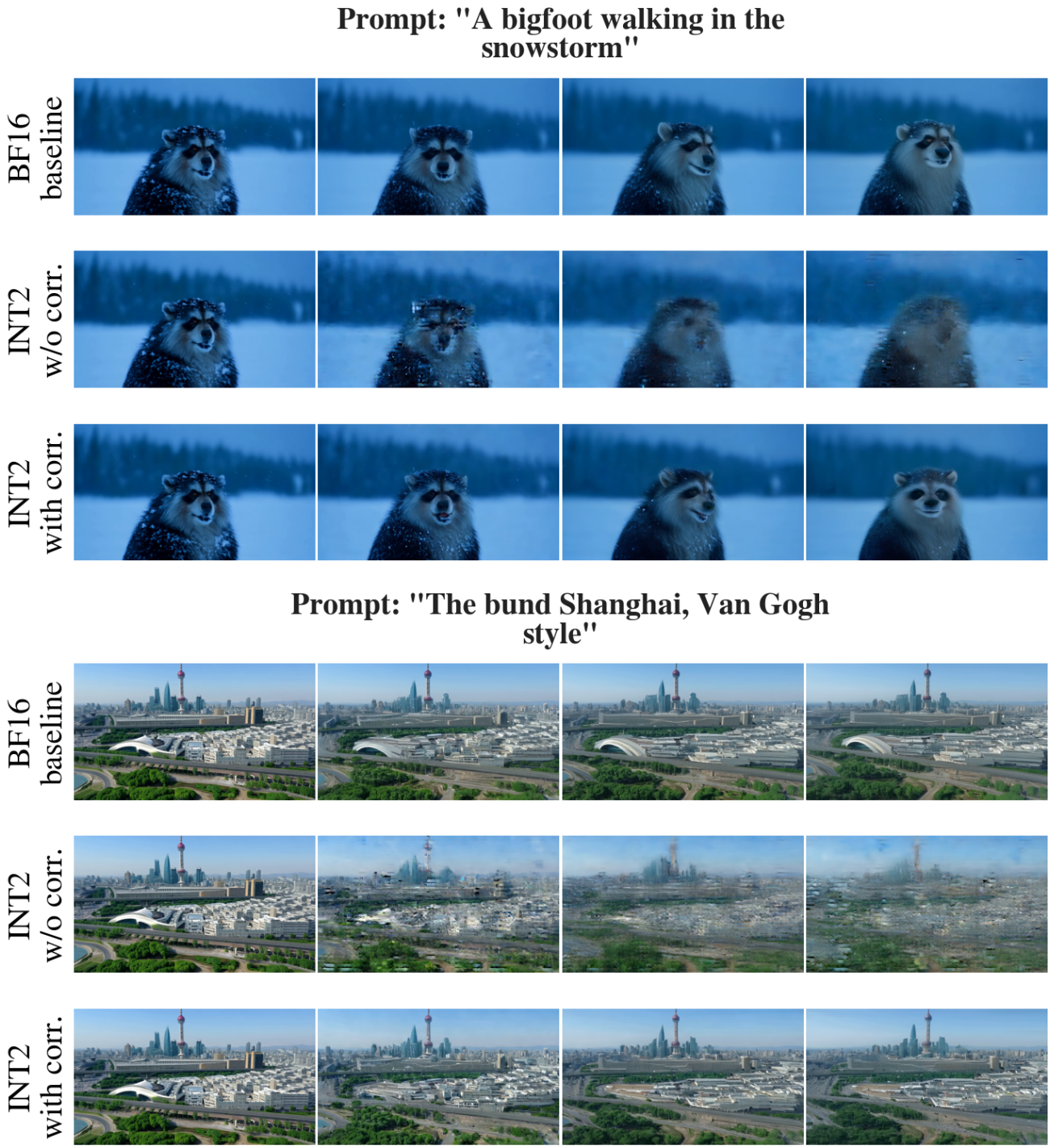


Figure 6. Qualitative comparison on SkyReels-V2. Columns show successive frames from the same video. Rows show BF16; INT2 asymmetric QuaRot+RTN quantization of cached keys and values; and the same setting with our correction. As on MAGI-1 (Figure 1), INT2 introduces visible distortions, while our correction recovers much of the BF16-like visual quality and temporal consistency.

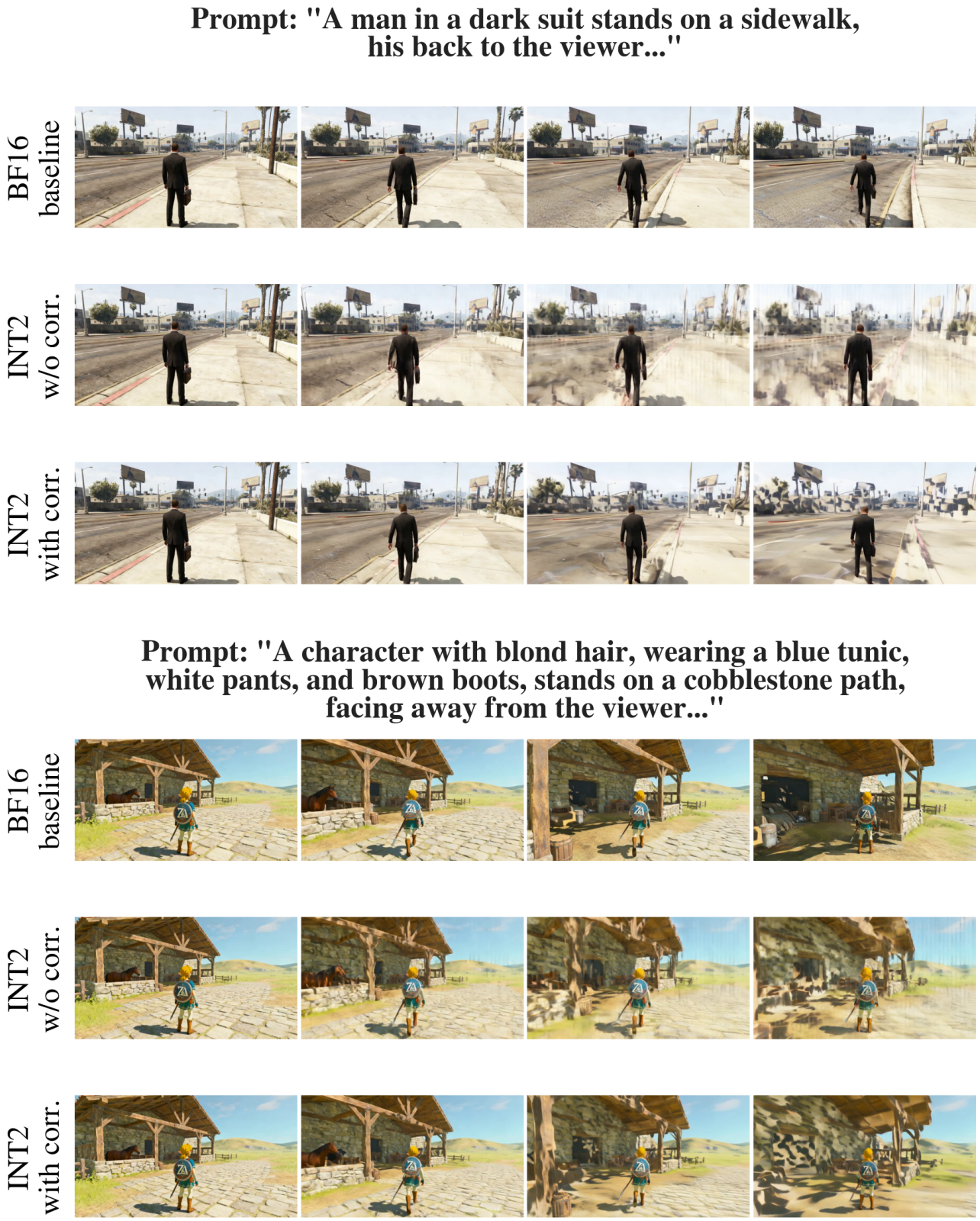


Figure 7. Qualitative comparison on HY-WorldPlay. Columns show successive frames from the same video. Rows show BF16; INT2 asymmetric QuaRot+RTN KV-cache quantization of keys and values; and the same quantized setting with our correction. As on MAGI-1 (Figure 1), INT2 introduces visible distortions, while our correction recovers much of the BF16-like visual quality and temporal consistency.

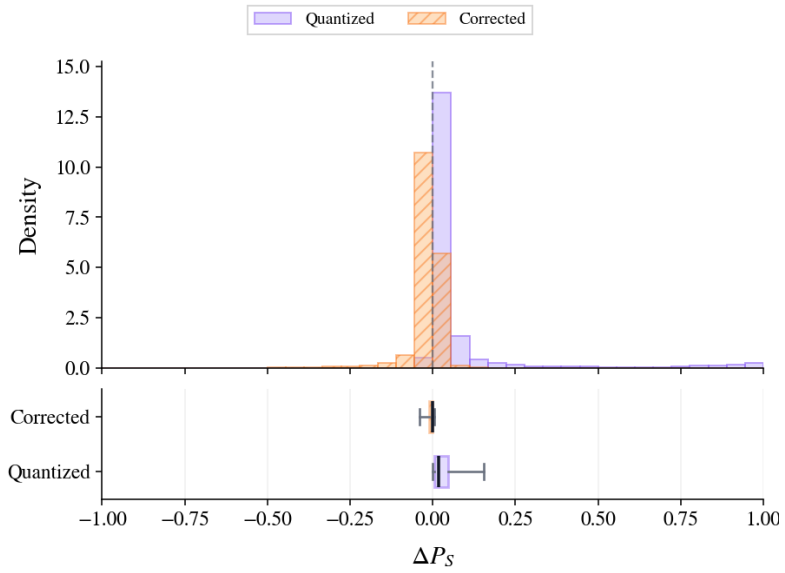
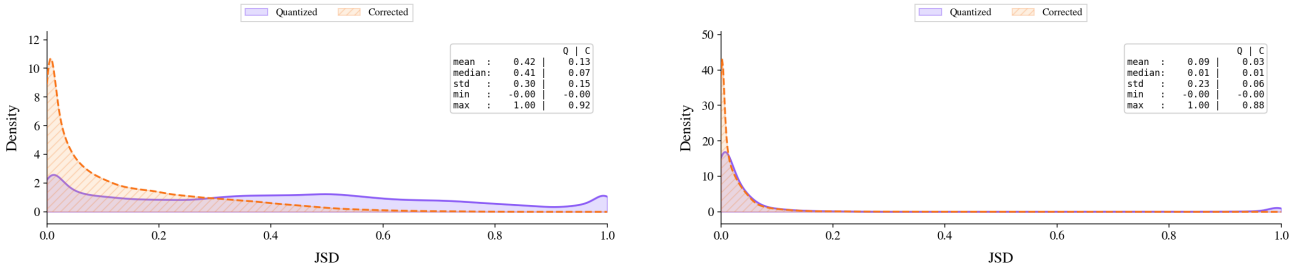
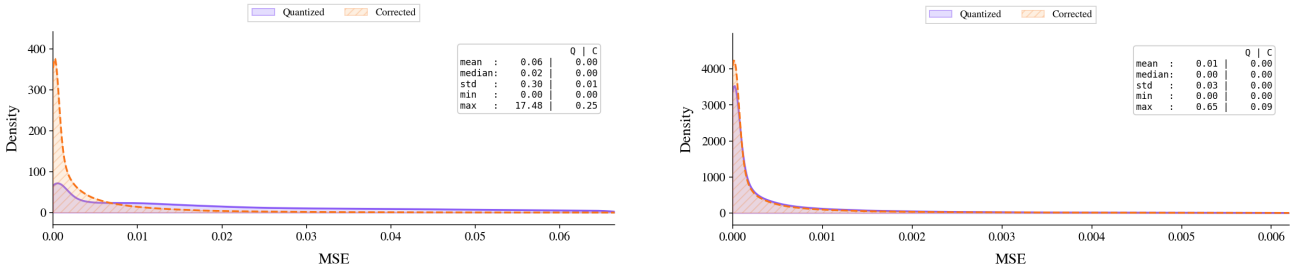


Figure 8. Cached attention mass shift ΔP_S on MAGI-1 with KV-cache quantization. Quantization shifts on average only $+0.08$ of attention mass onto the cached block (vs. $+0.37$ at INT2; cf. Figure 5). The correction centers the distribution near zero (mean -0.02).



(a) INT2 quantization *(b)* INT4 quantization

Figure 9. JSD between quantized/corrected and BF16 attention weights for INT2 and INT4.



(a) INT2 quantization *(b)* INT4 quantization

Figure 10. Attention output MSE for quantized/corrected for INT2 and INT4.

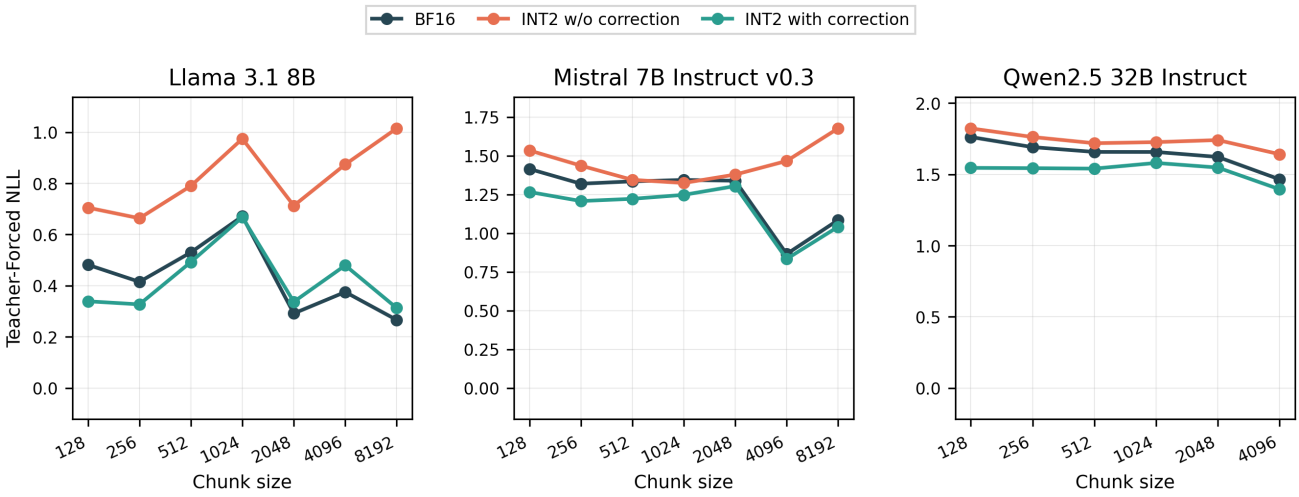


Figure 11. Teacher-forced NLL by partial-prefill chunk size in the LLM partial-prefill setting. Each panel corresponds to one model, and curves show BF16, plain INT2 KV-cache quantization, and INT2 with Taylor correction. Plain INT2 generally increases NLL, while the Taylor correction consistently reduces the degradation.

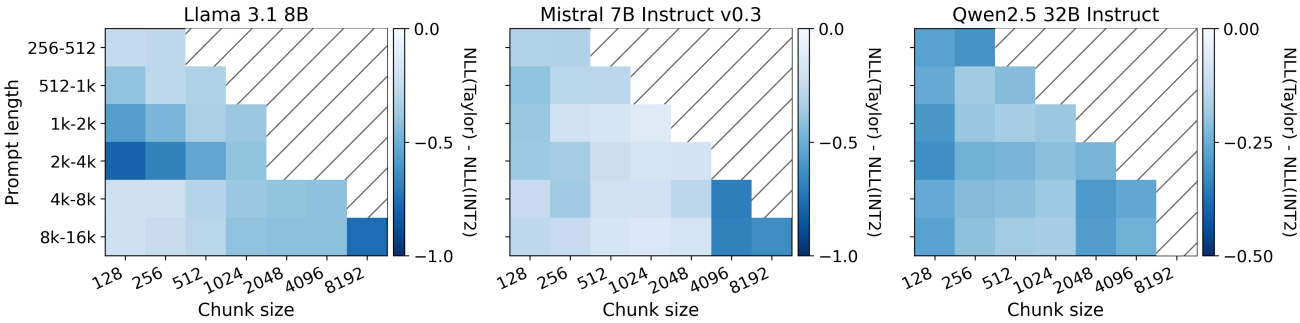


Figure 12. Prompt-length and chunk-size breakdown for LLM partial-prefill experiments. The plotted value is $\mathrm{NLL}_{\mathrm{INT2+Taylor}} - \mathrm{NLL}_{\mathrm{INT2}}$, computed within matched model, chunk-size, prompt-bin, and evaluation-example cells. Negative values indicate that the Taylor correction reduces teacher-forced NLL relative to plain INT2 KV-cache quantization. Striped areas indicate no available matched data.

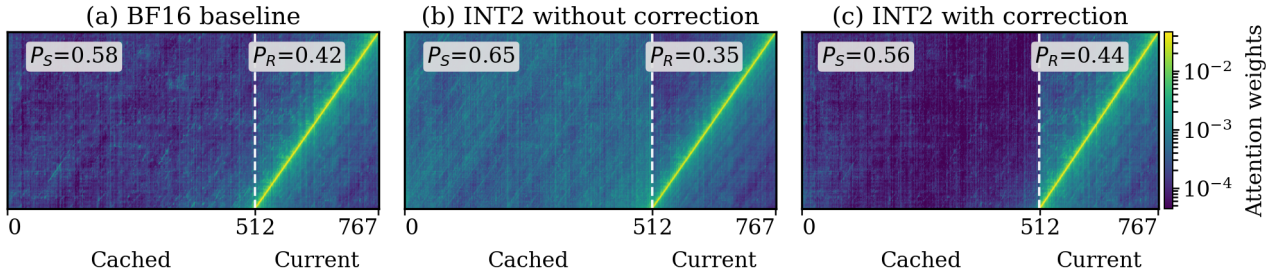


Figure 13. Attention weights for Llama-3.2-1B under INT2 KV-cache quantization. The visualized attention weights are averaged over representative prompts with lengths in $[1024, 2048)$, layers, and attention heads for chunk size 256. The dashed vertical line separates cached-prefix tokens from current-chunk 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.