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

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

Abstract

Autoregressive video diffusion models rely on a KV cache of previously generated frames to avoid redundant computation, but 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 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 tokens. We derive a per-attention-score correction that removes this bias in expectation, computed on the fly from quantization step sizes and query statistics; a second-order Taylor approximation makes the correction cheap and requires no additional KV-cache storage. Evaluated on MAGI-1, SkyReels-V2, and HY-WorldPlay at INT2 quantization, our correction improves PSNR by up to 5.9 dB and raises VBench Score by 7.8 points, recovering much of the quality lost to quantization.

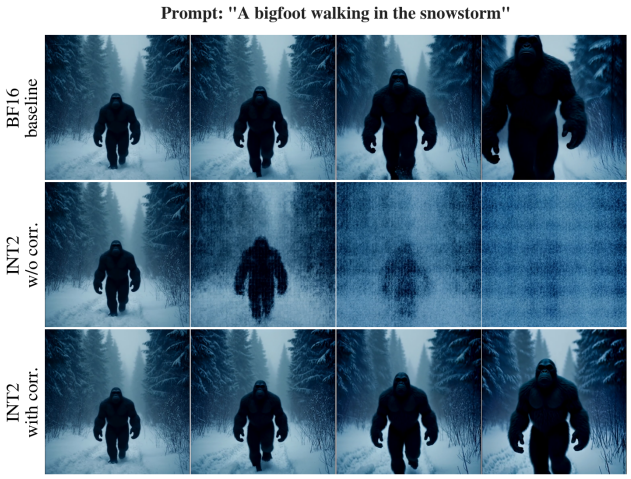


Figure 1. Qualitative comparison on MAGI-1. Columns show successive frames from the same generated video. INT2 QuaRot+RTN KV-cache quantization destroys subject and scene structure, while our correction substantially restores BF16-like visual quality and temporal consistency.

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 autoregressive video diffusion models extend this setting by denoising chunks of frames 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). This cache acts as temporal memory, determining the available past visual context when generating the next chunk.

Because the cache grows with the number of retained chunks, long-form generation faces a memory–context trade-off: larger windows improve temporal consistency but increase KV-cache memory proportionally. Low-bit KV-cache quantization can relax this trade-off by storing more past context under the same memory budget. While prior work has shown that LLM KV caches can be compressed to very low bitwidths (Liu et al., 2024; Hooper et al., 2024; Ashkboos et al., 2024), we find that applying INT2 KV-cache quantization to autoregressive video models severely distorts generated frames (Figure 1, Figure 5, Figure 6, Figure 7).

We trace this degradation to an attention-mass shift in chunk-wise video generation toward cached tokens under aggressive key quantization (Figure 3). Quantizing cached keys perturbs their attention scores, while current-chunk scores remain exact. Although this perturbation is approximately zero-mean before softmax, exponentiation is asymmetric: positive score errors increase e^{s_i} more than equal negative errors decrease it. The cached partition sum is therefore inflated in expectation, causing quantized cached keys to

[†]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 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

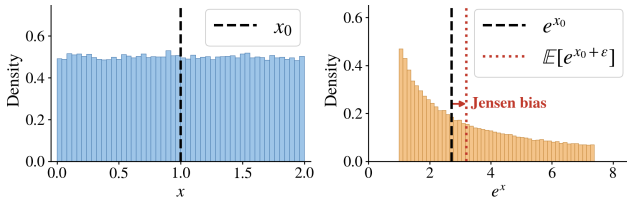


Figure 2. Illustration of the *Jensen bias*. Left: Zero-mean quantization noise keeps noisy scores centered around x_0. Right: After exponentiation, positive score errors are amplified more than negative errors are suppressed, so $\mathbb{E}[e^{x_0+\varepsilon}] > e^{x_0}$.

steal attention mass from the current chunk. We call this effect the *Jensen bias*.

We introduce a training-free correction that subtracts an estimated bias from cached-key attention scores before softmax. The correction is computed on the fly from existing quantization statistics at attention time and restores the attention mass balance between cached and current tokens.

Across MAGI-1, SkyReels-V2, and HY-WorldPlay, this simple correction recovers much of the quality lost to INT2 KV-cache quantization, improving both fidelity metrics and perceptual video quality while adding no storage overhead.

2. Related Work

Low-bit KV-cache quantization for LLMs typically targets channel-wise outliers in keys via per-channel treatment (Liu et al., 2024; Hooper et al., 2024), randomized rotations (Ashkboos et al., 2024; Zandieh et al., 2025), or asymmetric bit allocation between keys and values (Tao et al., 2024). Our work is orthogonal: rather than improving the representation, we analytically correct the bias any such scheme induces in the softmax partition sum.

Several works study how quantization perturbs attention. Closest to ours is KVLinC (Saxena & Roy, 2025), which compensates errors from quantized keys via *trainable* linear adapters; our correction, in contrast, is training-free and analytically derived. Other efforts target different sources of error: bias from quantizing the softmax itself (Pandey et al., 2023) or quantization-friendliness of the $QK^\top$ product (Zhang et al., 2025), neither of which addresses the convexity-induced inflation we identify.

Chunk-wise autoregressive video diffusion (Chen et al., 2024; Yin et al., 2025; Sand.ai et al., 2025; Sun et al., 2025) generates videos by denoising successively, attending to a KV cache of past chunks. Since this cache grows with video length, recent work reduces its cost through compression and eviction (Ma et al., 2026; Chen et al., 2026a; Samuel et al., 2026), sparse attention (Lv et al., 2026), or low-bit quantization (Xi et al., 2026). QuantVideoGen is closest to our setting: it attacks the quantization error via semantic-aware smoothing and residual quantization, whereas we leave the error in place and correct the bias it induces in softmax attention. We confirm this complementarity empirically: composing the two methods on MAGI-1 yields our best overall results (Table 1).

3. Preliminaries

Integer quantization maps a real value x to a discrete grid defined by a scale Δ and zero-point z:

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

Reconstruction $\hat{x} = (x_q - z)\cdot\Delta$ introduces the bounded error $\epsilon = \hat{x} - x$ with $|\epsilon| \le \Delta/2$.

The quantization parameters (Δ, z) can be shared at different granularities: per-tensor, per-token, or group-wise per-token with groups of size g. Finer granularity reduces quantization error at the cost of increased storage.

Optionally, a Hadamard rotation is applied prior to quantization to improve robustness (Ashkboos et al., 2024).

In autoregressive video diffusion, each chunk is represented as a set of spatio-temporal tokens. At each denoising step, queries from the current chunk attend to current keys (computed in full precision) and cached keys from previous chunks stored in the KV cache. The resulting attention matrix decomposes into a *current* and a *cached* block.

4. Method

We characterize how quantization induces a Jensen bias in softmax attention, and derive a practical correction.

4.1. Score-Space Noise Under Quantization

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

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

Let $\mathcal{S}$ and $\mathcal{R}$ denote *cached* (quantized) and *current-chunk* (unquantized) key indices, respectively. We define the partition sums and the cached attention mass

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

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 statistical

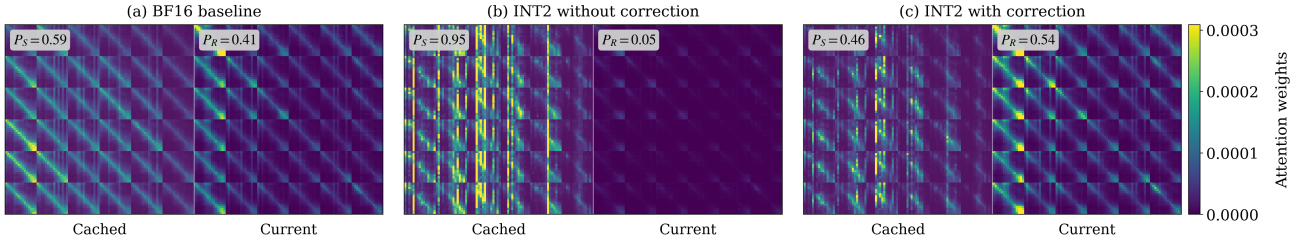


Figure 3. Attention weights for MAGI-1 for the prompt "a person" under INT2 KV-cache quantization. The visualization is taken from a representative layer, timestep, and attention head. Panel (b) shows that relative to the BF16 baseline in (a), quantization increases attention weights in the cached block 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.

model for round-to-nearest uniform quantization (Widrow et al., 1996), we assume independent channel-wise noise $\epsilon_{i,c} \sim \mathcal{U}(-\Delta_{i,c}/2, +\Delta_{i,c}/2)$. This noise model is approximate: real rounding errors can be signal-dependent or clipped, so we use it to predict the bias direction and validate the correction empirically under the actual QuaRot+RTN implementation.

The quantized attention score is therefore

$$\hat{s}_i = \frac{q^\top \hat{k}_i}{\sqrt{d}} = s_i + \delta_i, \qquad \delta_i = \frac{q^\top \epsilon_i}{\sqrt{d}}. \tag{4}$$

Thus δ_i is zero-mean under the model, while unquantized current keys satisfy $\hat{s}_i = s_i$.

4.2. Quantization Bias in Softmax Attention

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

$$\mathbb{E}\big[\hat{Z}_{\mathcal{S}}\big] = \sum_{i \in \mathcal{S}} e^{s_i} \cdot \mathbb{E}\big[e^{\delta_i}\big]. \tag{5}$$

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

$$\Delta P_{\mathcal{S}} = \hat{P}_{\mathcal{S}} - P_{\mathcal{S}}, \qquad \hat{P}_{\mathcal{S}} = \frac{\hat{Z}_{\mathcal{S}}}{\hat{Z}_{\mathcal{S}} + Z_{\mathcal{R}}}. \tag{6}$$

Positive values indicate excess attention on cached tokens (Section 5.3).

4.3. Correction of the Jensen Bias

We apply a per-score correction b_i only to cached scores ($i \in \mathcal{S}$) such that each token's contribution to the partition sum is unbiased in expectation, $\mathbb{E}[e^{s_i - b_i + \delta_i}] = e^{s_i}$. This yields the unique solution:

$$e^{s_i - b_i} \cdot \mathbb{E}\big[e^{\delta_i}\big] = e^{s_i} \quad \Longrightarrow \quad \boxed{b_i = \log \mathbb{E}\big[e^{\delta_i}\big].} \tag{7}$$

At inference time, we apply this correction by subtracting b_i from each cached attention score 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), so the correction always subtracts from cached scores. Furthermore, b_i increases with the score-space noise, i.e. with $\Delta_{i,c}$.

Since the noise components $\epsilon_{i,c}$ are independent across channels and each $\epsilon_{i,c} \sim \mathcal{U}(-\Delta_{i,c}/2,\ +\Delta_{i,c}/2)$, the expectation $\mathbb{E}[e^{\delta_i}]$ factorizes across dimensions. We defer the exact expression to Section A and derive a second-order approximation suitable for efficient implementation based on $\alpha_c = q_c \Delta_{i,c}/(2\sqrt{d})$. Given $\log(\sinh(\alpha)/\alpha) \approx \alpha^2/6$ for small $|\alpha_c|$, we obtain

$$\boxed{b_i = \sum_{c=1}^{d} \log\left(\frac{\sinh\left(\dfrac{q_c\,\Delta_{i,c}}{2\sqrt{d}}\right)}{\dfrac{q_c\,\Delta_{i,c}}{2\sqrt{d}}}\right) \overset{\text{Taylor}}{\approx} \frac{1}{24\,d} \sum_{c=1}^{d} q_c^2\,\Delta_{i,c}^2} \tag{8}$$

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

For group-wise quantization with $G = d/g$ groups sharing step size $\Delta_{i,j}$, $\|q_j\|^2$ denoting the squared norm of q restricted to group j, Equation (8) simplifies to

$$b_i \approx \frac{1}{24\,d} \sum_{j=1}^{G} \Delta_{i,j}^2\,\|q_j\|^2. \tag{9}$$

The extension to QuaRot replaces q with the rotated query Hq (see Section F).

Table 1. Effect of the proposed correction on benchmark results 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. Rows marked INT2 (QVG) apply QuantVideoGen compression (Xi et al., 2026) on MAGI-1.

Model	Prec.	Corr.	PSNR↑	SSIM↑	LPIPS↓	VBench↑
MAGI-1	BF16		–	–	–	78.27
	INT2	×	17.10	0.630	0.453	70.24
		✓	**22.97**	**0.801**	**0.165**	**78.02**
	INT2 (QVG)	×	23.01	0.826	0.132	77.81
		✓	**25.29**	**0.856**	**0.107**	**78.23**
SkyReels-V2	BF16		–	–	–	78.89
	INT2	×	19.20	0.708	0.319	71.44
		✓	**20.42**	**0.784**	**0.202**	**78.58**
HYWP	INT2	×	17.16	0.575	0.376	–
		✓	**18.27**	**0.616**	**0.273**	–

4.4. Compute and Memory Overhead.

The correction adds no storage (it reuses existing per-group scales) and only $O(QK \cdot d/g)$ additional compute, a factor g smaller than $QK^\top$. With $B = 2$ and $g = 32$, the effective bitwidth is $B_{\text{eff}} = B + 24/g = 2.75$. See Section B for the full breakdown.

5. Experiments

5.1. Experimental Setup

We evaluate three chunk-wise 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). Unless noted, cached keys and values use group-wise per-token asymmetric INT2 QuaRot+RTN (Ashkboos et al., 2024) with $g = 32$, FP8 E4M3 scales, and BF16 zero-points. 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. Our Taylor correction targets only the key-induced softmax bias; value quantization is unchanged.

We use BF16 generations as references for PSNR, SSIM, and LPIPS (Zhang et al., 2018). For MAGI-1 and SkyReels-V2, we report aggregate VBench-Long Scores (Huang et al., 2023). For HY-WorldPlay, we use the 10 released image–prompt pairs and report only fidelity metrics because VBench lacks the required action inputs.

5.2. Main Results

KV-cache quantization substantially degrades all three models (Table 1, Figure 1), consistent with the predicted attention-mass shift. Without additional storage cost, our correction improves all reported fidelity metrics and nearly restores VBench: from 70.24 to 78.02 on MAGI-1, close to the BF16 baseline of 78.27, and from 71.44 to 78.58 on SkyReels-V2, see also the full breakdown including standard errors in Section G. HY-WorldPlay shows smaller but consistent gains, suggesting that Jensen bias is present across architectures, with model-dependent magnitude. 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.

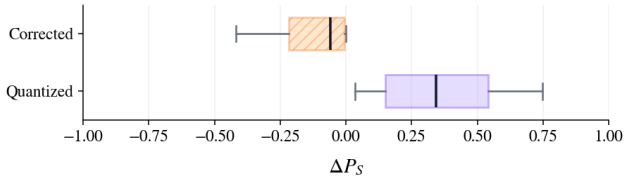


Figure 4. Cached attention-mass shift on MAGI-1 under INT2 KV-cache quantization. Positive values indicate attention stealing by quantized cached keys. Our correction substantially centers the distribution toward zero.

5.3. Analysis

We validate the mechanism on MAGI-1 by measuring the cached attention-mass shift $\Delta P_S = \hat{P}_S - P_S$, averaged over prompts, heads, layers, and denoising steps. Quantization strongly shifts mass to cached tokens (Figure 4); the correction moves the distribution back toward zero, with slight Taylor-induced over-correction. Attention JSD and output MSE also improve (see Sections L and M).

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 identify the *Jensen bias* as a systematic failure mode of low-bit KV-cache quantization: quantized cached keys re-

ceive inflated softmax mass and steal attention from current tokens. We derive a training-free correction and show that its Taylor approximation recovers most of the lost quality with negligible overhead across three architectures spanning 1.3B to 8B parameters.

Limitations. Our study focuses on chunked autoregressive video diffusion. While the same Jensen bias appears in quantized LLM KV caches, standard single-token decoding offers limited headroom for correction, as cached tokens compete with only one unquantized token.

The correction removes bias only in expectation and works best when attention is distributed over many cached tokens, as in long-context video generation. For highly peaked attention, individual noise realizations can dominate and limit 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 is a post-hoc correction for trained models and does not weaken safety mechanisms. By making existing video generation more efficient, it may lower deployment costs and inherits the dual-use risks of underlying models.

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.

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.

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.

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.

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.

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.

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. The notation $\Delta_{i,c}$ denotes the quantization step size associated with channel c of cached token i; different quantization granularities correspond to different sharing patterns among these step sizes.

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:

$$\mathbb{E}\big[e^{\delta_i}\big] = \prod_{c=1}^{d} \mathbb{E}\left[\exp\left(\frac{q_c\, \epsilon_{i,c}}{\sqrt{d}}\right)\right]. \tag{10}$$

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

$$\begin{aligned}\mathbb{E}\big[e^{t_c\, \epsilon_{i,c}}\big] &= \frac{1}{\Delta_{i,c}} \int_{-\Delta_{i,c}/2}^{+\Delta_{i,c}/2} e^{t_c\, u}\, du \\ &= \frac{\sinh(t_c\, \Delta_{i,c}/2)}{t_c\, \Delta_{i,c}/2}.\end{aligned} \tag{11}$$

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

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

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.

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:

$$b_i \approx \sum_{c=1}^{d} \frac{\alpha_c^2}{6} = \frac{1}{24\, d} \sum_{c=1}^{d} q_c^2\, \Delta_{i,c}^2. \tag{13}$$

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 reduces to

$$b_i \approx \frac{1}{24\, d} \sum_{j=1}^{G} \Delta_{i,j}^2\, \|q_j\|^2, \tag{14}$$

where $\|q_j\|^2 = \sum_{c \in \mathcal{G}_j} q_c^2$.

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

$$B_{\text{eff}} = \frac{d \cdot B + 24 \cdot G}{d} = B + \frac{24}{g}. \tag{15}$$

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.

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:

$$b = \sum_{c=1}^{d} \log \left(\frac{\sinh\left(\dfrac{q_c\, \Delta_c}{2\sqrt{d}}\right)}{\dfrac{q_c\, \Delta_c}{2\sqrt{d}}} \right), \tag{16}$$

with the Taylor approximation

$$b \approx \frac{1}{24\, d} \sum_{c=1}^{d} q_c^2\, \Delta_c^2. \tag{17}$$

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.

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

$$\hat{s}_i = \frac{(Hq)^\top (Hk_i + \epsilon_i)}{\sqrt{d}} = s_i + \delta_i^{(H)}, \tag{18}$$

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

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

$$b_i^{(H)} \approx \frac{1}{24\, d} \sum_{j=1}^{G} \Delta_{i,j}^2\, \|(Hq)_j\|^2. \tag{19}$$

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. Per-Dimension VBench Results

Table 1 reports the aggregate VBench Score on MAGI-1 and SkyReels-V2. For completeness, Tables 2 and 3 break this score down across all 16 VBench dimensions, grouped by VBench's *Quality* (visual fidelity) and *Semantic* (prompt fidelity) categories, and Table 4 reports the corresponding sub-scores together with the Total VBench Score that already appears in Table 1.

H. Additional Qualitative Comparison on MAGI-1

Figure 1 in the main text shows the qualitative effect of INT2 KV-cache quantization and our correction on MAGI-1 for a single prompt. Figure 5 provides two additional examples on different prompts, confirming that the same pattern holds: uncorrected INT2 QuaRot+RTN quantization severely degrades subject and scene structure, while our correction substantially recovers BF16-like visual quality and temporal consistency.

Table 2. 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 3. 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

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.

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 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 Figure 8 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,

Table 4. 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

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.

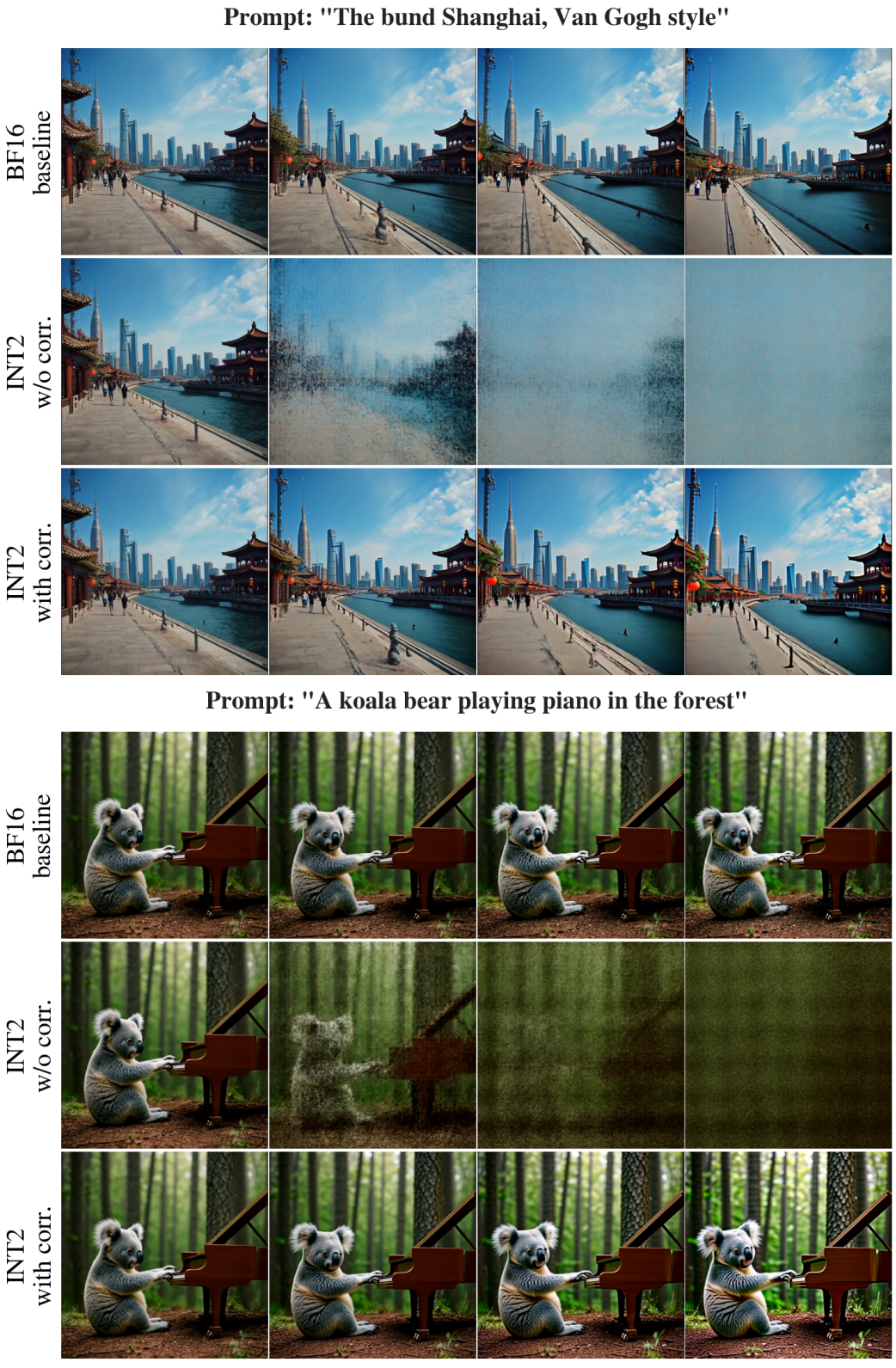


Figure 5. Additional qualitative comparisons on MAGI-1 for two different prompts. Columns show successive frames from the same generated video. From top to bottom in each panel: BF16 baseline; INT2 asymmetric QuaRot+RTN KV-cache quantization of both keys and values; same quantized setting with our correction.

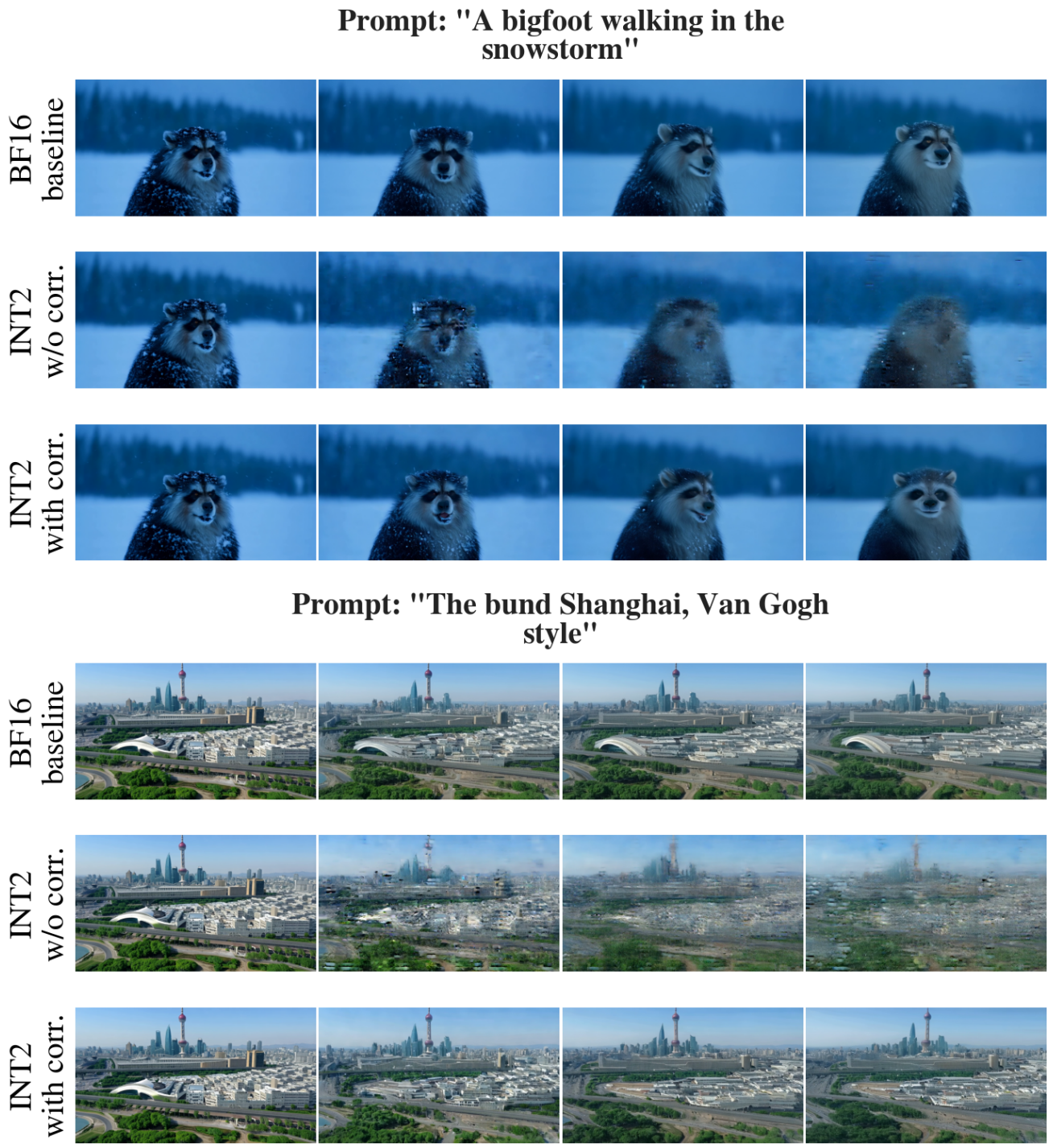


Figure 6. Qualitative comparison on SkyReels-V2. Columns show successive frames from the same video. Rows show BF16; INT2 asymmetric QuaRot+RTN quantization of cached keys and values; and the same setting with our correction. As on MAGI-1 (Figure 1), INT2 introduces visible distortions, while our correction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

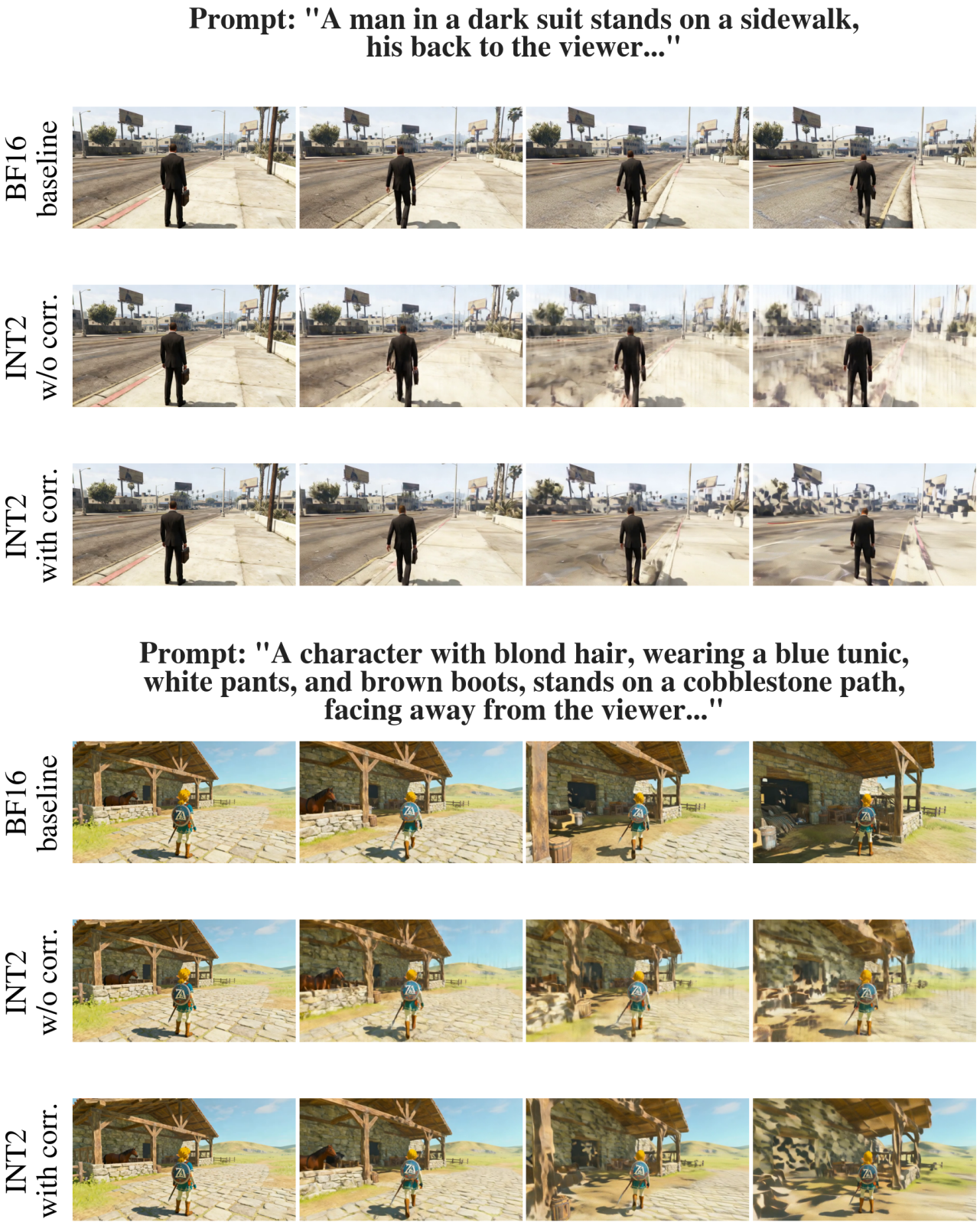


Figure 7. Qualitative comparison on HY-WorldPlay. Columns show successive frames from the same video. Rows show BF16; INT2 asymmetric QuaRot+RTN KV-cache quantization of keys and values; and the same quantized setting with our correction. As on MAGI-1 (Figure 1), INT2 introduces visible distortions,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

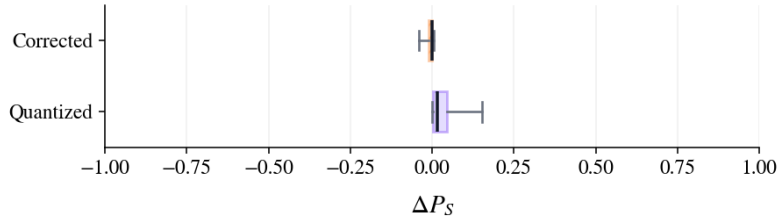
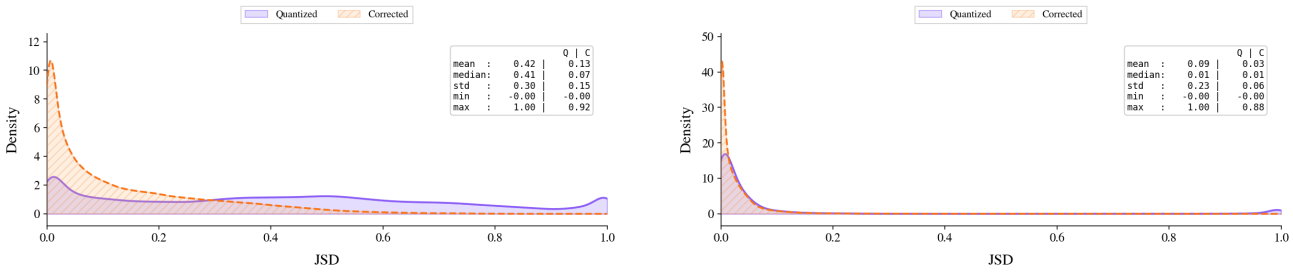
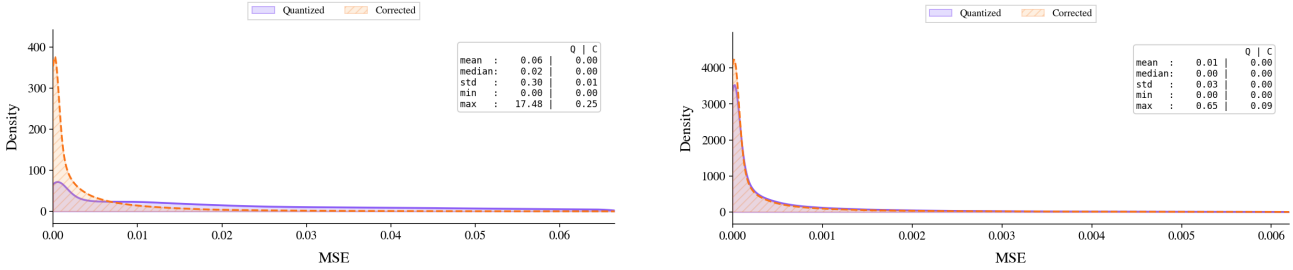


Figure 8. Cached attention mass shift ΔP_S on MAGI-1 with INT4 KV-cache quantization. Quantization shifts on average only $+0.08$ of attention mass onto the cached block (vs. $+0.37$ at INT2; cf. Figure 4). The correction centers the distribution near zero (mean -0.02).



(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.

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 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.

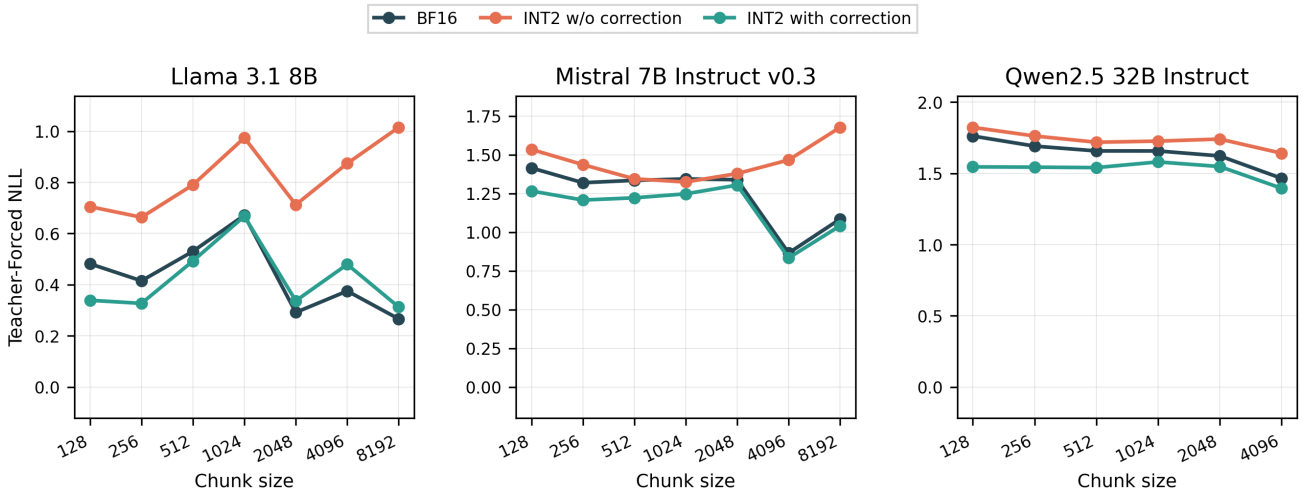


Figure 11. Teacher-forced NLL by partial-prefill chunk size in the LLM partial-prefill setting. Each panel corresponds to one model, and curves show BF16, plain INT2 KV-cache quantization, and INT2 with Taylor correction. Plain INT2 generally increases NLL, while the Taylor correction consistently reduces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

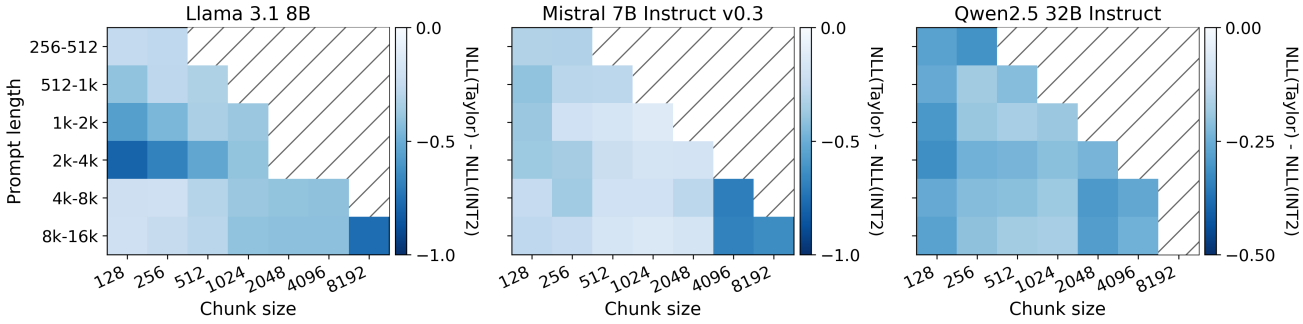


Figure 12. Prompt-length and chunk-size breakdown for LLM partial-prefill experiments. The plotted value is $\mathrm{NLL}_{\mathrm{INT2+Taylor}} - \mathrm{NLL}_{\mathrm{INT2}}$, computed within matched model, chunk-size, prompt-bin, and evaluation-example cells. Negative values indicate that the Taylor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

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

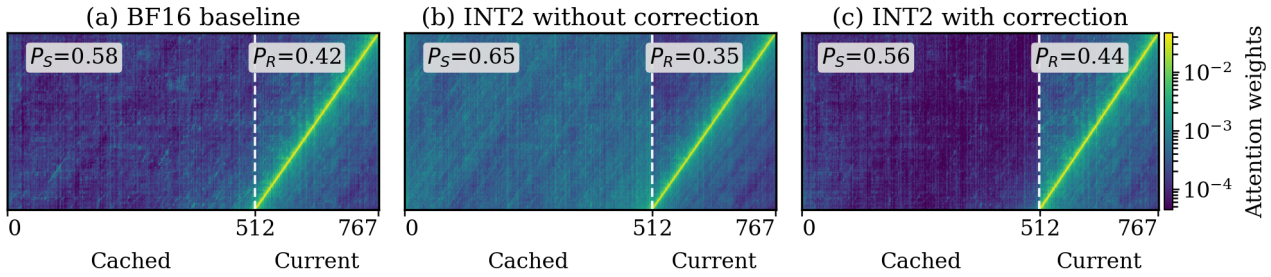


Figure 13. Attention weights for Llama-3.2-1B under INT2 KV-cache quantization. The visualized attention weights are averaged over representative prompts with lengths in $[1024, 2048)$, layers, and attention heads for chunk size 256. The dashed vertical line separates cached-prefix tokens from current-chunk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

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.