FREEATTN: UNLOCKING NATIVE E3M4 FOR EFFICIENT AND ACCURATE ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

The quadratic cost of attention limits long-context prefill and video generation. Low-bit quantization increases throughput but introduces numerical error. At the same eight-bit width, E3M4 trades a narrower dynamic range for finer normal-number spacing than E4M3. We independently identify and validate native packed `F2FP` conversion and matrix multiply–accumulate (MMA) modes for E3M4 on Blackwell SM120, and connect them into the native attention path of FREEATTN. The framework relates QK/PV quantization errors to attention outputs under shared scaling, organizes composable corrections through attention identities, and reduces their cost through shared statistics and fused execution. Matched-format controls show lower aggregate BF16-relative probability and pixel error with E3 than E4. Evaluated for both fidelity and latency, optimized all-FP8 E3 accelerates Qwen3-8B/32B prefill by 1.15–$1.16\times$ and Wan2.1-14B video delivery by $1.43\times$ over Torch SDPA. The FP4 path with Basic correction achieves 1.45–$1.51\times$ complete-attention-call speedup over Torch SDPA on seven common shapes, with lower aggregate operator output error than same-bitwidth SageAttention3. These results establish native E3M4 as a useful representation and execution option for attention inference.

1 INTRODUCTION

Attention is a major performance bottleneck in long-context language models and video generation. Its quadratic cost increases language-model prefill latency and recurs across the denoising steps of video diffusion models (Yang et al., 2025; Wan Team, 2025). Deployment systems commonly use quantization to reduce arithmetic and memory costs and increase serving throughput (Xiao et al., 2023). The resulting numerical errors propagate through the model, changing output distributions and generated content. Low-bit attention must therefore reduce inference cost while preserving the capabilities and output behavior established through fine-tuning and alignment.

Existing approaches reduce attention cost through efficient memory access and low-precision matrix products. FlashAttention uses tiling and online softmax to reduce memory traffic (Dao et al., 2022; Dao, 2024). Low-precision methods further accelerate the query–key product (QK) and probability–value product (PV) using FP8, integer, or FP4 tensor-core modes (Shah et al., 2024; Zhang et al., 2024b;a; 2025b;a). Error-control mechanisms reshape operand distributions before quantization: FlashAttention-3 spreads outliers through orthogonal transforms, while the SageAttention family removes shared offsets through centering or restores their score contributions through compensation. Finer quantization granularity in SageAttention2 and two-level attention-weight quantization in SageAttention3 further reduce representation error. These mechanisms remain constrained by the range–resolution tradeoff of the selected format; explicit compensation also adds preparation and arithmetic costs. The matched-prompt examples in Figure 1 show that different precision configurations still produce different content under identical prompts and initial noise.

Operand bitwidth alone does not determine low-precision attention fidelity. Group quantization assigns a common positive scale to a group of values: each value is divided by that scale and rounded to the low-precision format (Darvish Rouhani et al., 2023b). The shared scale accommodates magnitude differences between groups, while the element format determines representable range and spacing within a group. Format selection therefore affects accuracy even at a fixed bitwidth (Kuzmin et al., 2022). E3M4 assigns one fewer bit to the exponent and one more to the significand than E4M3,

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

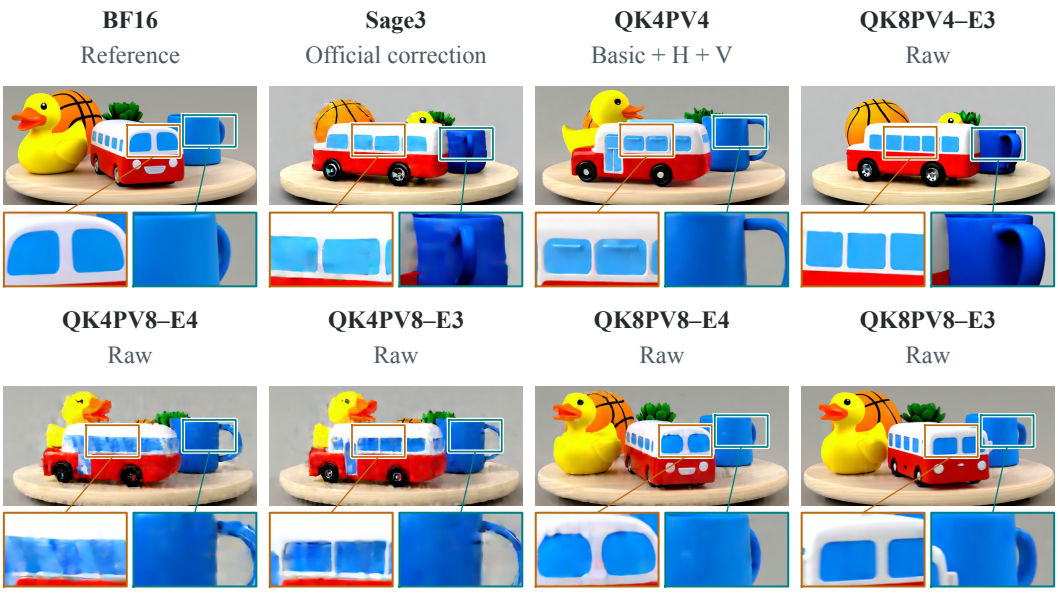


Figure 1: Wan14B outputs with identical prompt and initial noise: frame 27 and equal-scale detail crops. Correction states are labeled; QK4PV4 uses Smooth-Q/K/V and Hadamard. BF16 is the reference.

providing finer spacing over their common normal range at the cost of reduced small-value coverage. Efficient E3M4 execution requires native hardware support for both conversion and matrix computation.

Public ISA analyses describe E3M4 matrix execution (kacper-daftcode, 2026; NVIDIA, 2026c). We independently recover and connect Blackwell SM120's packed `F2FP` conversion and MMA modes for native E3M4 attention. The conventional PTX type interfaces do not directly expose these modes (NVIDIA, 2026b). Independent numerical references validate conversion semantics and matrix-operand semantics on SM120. Connecting native F2FP to MMA encodes online-softmax weights directly in registers for PV, avoiding software encoding inside the attention loop. Built on this path, FREEATTN combines numerical error analysis, composable corrections, and fused execution to improve the fidelity and efficiency of low-bit attention.

Whether E3M4's finer spacing compensates for its reduced small-value coverage depends on error propagation through attention. QK error changes attention weights through softmax; PV error changes how those weights combine value vectors. The output effect of a probability perturbation depends on the value vectors to which it is assigned; dominant QK error can also mask an improvement in PV. The analysis connects these effects to the range–resolution tradeoff under shared scaling. Reducing operand or probability error is therefore insufficient to guarantee lower attention-output error. To control these effects, FREEATTN organizes equivalent transforms, explicit compensation, and local protection into a composable correction menu. Building on existing centering and smoothing methods, the framework establishes their composition conditions under online softmax and reduces execution cost by sharing statistics and fusing compatible operations.

In the matched-format language and video evaluations, all-FP8 E3 improves aggregate BF16-relative probability and pixel fidelity over matched E4. Separate evaluations of the optimized all-FP8 E3 path measure fidelity and latency on the same model inputs, with 1.15–$1.16\times$ faster Qwen3-8B/32B prefill and $1.43\times$ faster Wan2.1-14B video delivery than Torch SDPA. The same correction and fusion framework applies to FP4: Basic FP4 achieves 1.45–$1.51\times$ complete-call speedup over Torch SDPA on seven common shapes, with lower aggregate operator output error than same-bitwidth SageAttention3. Correction ablations also show that lower operator error does not necessarily imply higher model-output fidelity. These results support choosing precision according to fidelity requirements rather than minimizing bitwidth alone.

The contributions are threefold:

- **Native E3M4 conversion and MMA for attention.** FREEATTN connects independently recovered E3M4 F2FP and MMA modes through consistent scaling and packing conventions. Independent numerical references validate the two instruction interfaces, and an optimized software-encoding control isolates the benefit of native conversion (Section 3.2).
- **Numerical representation and attention-error propagation.** Shared-scale format analysis characterizes the range–resolution tradeoff. An output-error decomposition relates QK perturbations to value vectors and shows how QK error can mask PV improvements, explaining workload-dependent asymmetric-precision choices (Section 3.3).
- **Composable corrections and efficient fused execution.** The correction framework specifies cancellation, compensation, and normalization conditions under online softmax and quantized weights. Shared statistics and fused preparation and recovery reduce execution cost. Operator and model experiments characterize the fidelity–cost tradeoff, including cases where additional correction does not improve generation fidelity (Sections 3.4–3.5).

2 BACKGROUND

Attention connects two matrix products through rowwise softmax:

$$S = QK^\top/\sqrt{D} + M, \qquad P = \mathrm{softmax}_{\mathrm{row}}(S), \qquad O = PV, \tag{1}$$

where $Q \in \mathbb{R}^{N_q \times D}$, $K, V \in \mathbb{R}^{N_k \times D}$, and M masks invisible positions with $-\infty$ (Vaswani et al., 2017). QK determines relative scores; PV aggregates the values using the resulting weights. Their quantization errors therefore enter the output through different paths.

FlashAttention evaluates these products with online softmax without materializing the full score or weight matrices (Milakov & Gimelshein, 2018; Dao et al., 2022). For each KV block, it updates a running row maximum m, forms $E = \exp(S - m)$, and updates the denominator ℓ and weighted accumulator U. When m increases, both accumulators are rescaled; the final output is U/ℓ. Low-precision execution must encode the newly generated E inside this loop, as well as prepare Q/K/V.

Group quantization reconstructs x_i as $\widehat{x}_i = \alpha_b \mathcal{Q}_f(x_i/\alpha_b)$, with positive scale α_b shared within group b (Darvish Rouhani et al., 2023b). The scale adapts to group magnitude, while format f determines within-group range and spacing. Efficient execution requires agreement between conversion, scale grouping, and matrix-operand interpretation. Centering, inverse Q/K scaling, and orthogonal transforms reshape values before quantization (Zhang et al., 2024a; Xiao et al., 2023; Ashkboos et al., 2024); their attention-specific cancellation and recovery conditions are developed in Section 3.4.

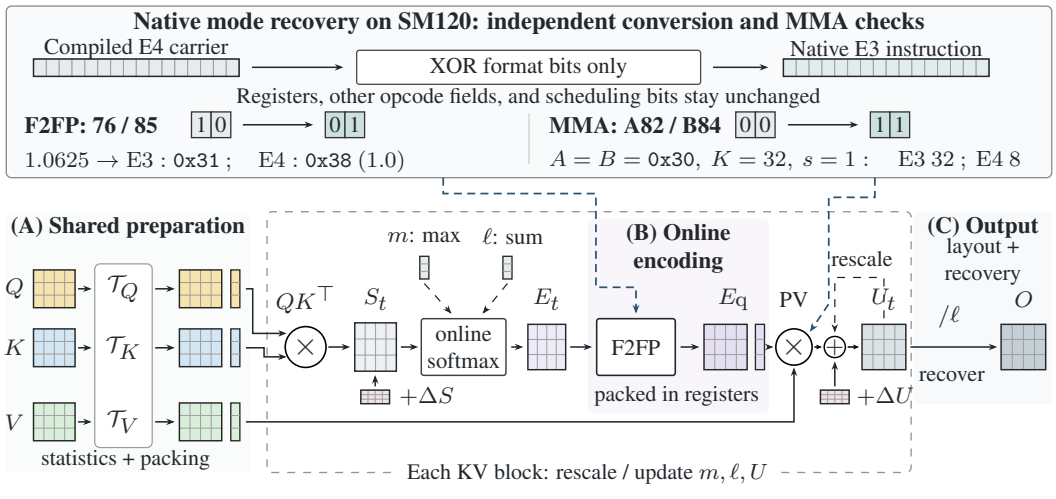
3 DESIGN AND ANALYSIS OF FREEATTN

3.1 OVERVIEW

FREEATTN connects numerical representation to native streaming execution (Figure 2). Input transforms reshape Q/K/V before encoding. Score compensation ΔS enters before softmax; native F2FP packs its newly generated weights for PV. Local protection contributes ΔU to the same online accumulator, and output recovery respects the effective weight row sum. QK and PV can independently use FP4 or FP8; QKxPVy denotes their operand bitwidths, without constraining softmax or accumulation precision.

3.2 UNLOCKING NATIVE E3M4

Identify the two interfaces independently. Controlled compilations that differ only in operand format isolate candidate fields in Blackwell instruction words (Jia et al., 2018; Jarmusch et al., 2025). The public `cvt.rn.satfinite.e4m3x2.f32` instruction provides a packed-conversion carrier (NVIDIA, 2026b). Enumerating candidate selectors and testing an independent E3M4 reference identifies its native mode. The matrix interface is recovered separately: conversion correctness alone cannot establish how MMA interprets the resulting bytes.



Schematic blocks; payload-adjacent strips denote scales. $E_t = \exp(S_t - m)$; S_t, E_t are block-local.

Figure 2: Native E3M4 attention. The inset shows independent converter/MMA recovery; the dataflow connects input preparation (A), online encoding (B), and output recovery (C). Compensation enters at the corresponding score, accumulator, or output stage.

For the SM120 instructions studied here, an E4M3 carrier w_4 is rewritten in place as

$$w_3 = w_4 \oplus \mathcal{M}, \qquad \mathcal{M}_{\mathrm{F2FP}} = 2^{76} + 2^{85}, \quad \mathcal{M}_{\mathrm{MMA}} = 2^{82} + 2^{84}. \tag{2}$$

Bits are numbered from the least significant bit of a 128-bit word. The MMA mask changes both operands; either bit can select one independently. Scoped binary differences verify that only identified format bits change, retaining code layout and scheduling. Conversion tests cover rounding, saturation, and packing boundaries; matrix tests independently enumerate all 256 byte encodings of each operand, including expected NaNs. Appendix A gives extraction details and architectural scope.

Connect representation to execution. Scales must use the selected format's range. For a nonzero group,

$$s_b = \left\lceil \log_2 \frac{\max_{i \in b} |x_i|}{F_f} \right\rceil, \qquad \widehat{x}_i = 2^{s_b} \mathcal{Q}_f(x_i 2^{-s_b}), \tag{3}$$

where $F_{\mathrm{E3M4}} = 30$ and $F_{\mathrm{E4M3}} = 448$. Matching scales and packing to MMA allows F2FP to encode online-softmax weights directly in registers. This matters because attention generates BHN^2 logical weights from $3BHND$ input elements: online encoding grows relative to input preparation as $N/(3D)$. Section 4.3 isolates this benefit against optimized software encoding at fixed E3 arithmetic.

3.3 Representation and Error Propagation

E3M4 exchanges one exponent bit for one significand bit, halving spacing in the common normal range while reducing small-value coverage. With group maximum 1, Equation 3 gives scales $1/16$ and $1/256$ for E3M4 and E4M3; their smallest positive reconstructed values are 2^{-10} and 2^{-17}. The finer resolution can thus coexist with additional zeros. Its value depends on how those perturbations affect attention.

Let P_{QK} be the normalized weights after QK quantization but before weight encoding, and $\widehat{O}$ the low-precision output. Define $e_{\mathrm{QK}} = (P_{\mathrm{QK}} - P)V$ and $e_{\mathrm{PV}} = \widehat{O} - P_{\mathrm{QK}}V$. Their sum is the output error, giving the exact identity

$$\|\widehat{O} - PV\|_F^2 = \|e_{\mathrm{QK}}\|_F^2 + \|e_{\mathrm{PV}}\|_F^2 + 2\langle e_{\mathrm{QK}}, e_{\mathrm{PV}} \rangle_F. \tag{4}$$

The PV term includes weight encoding, value quantization, and accumulation. The cross term completes the identity but need not dominate in practice. With Q/K fixed, additional PV precision cannot change the propagated component e_{QK}. In the controls of Appendix B.3, dominant QK error nearly masks an isolated PV improvement.

The values also matter. For one reference row o, write $\delta p_j = P_{\mathrm{QK},j} - P_j$. Since both probability rows sum to one, $e_{\mathrm{QK}} = \sum_j \delta p_j (v_j - o)$. Equal-sized probability perturbations in different directions can therefore have different output effects. Asymmetric precision should therefore be evaluated through propagated output error on identical inputs. Section 4.2 tests QK4PV8 and QK8PV4 and finds workload-dependent orderings.

3.4 Composable Error Correction

Centering, scaling, rotation, and selective higher precision are established quantization tools (Zhang et al., 2024a; Xiao et al., 2023; Ashkboos et al., 2024; Dettmers et al., 2022). FreeAttn organizes them by *where their effects cancel or must be restored*, then makes the composition consistent with online softmax and weight quantization. Figure 3 maps the nine corrections to their computation stages; Appendix C gives the complete composition rules.

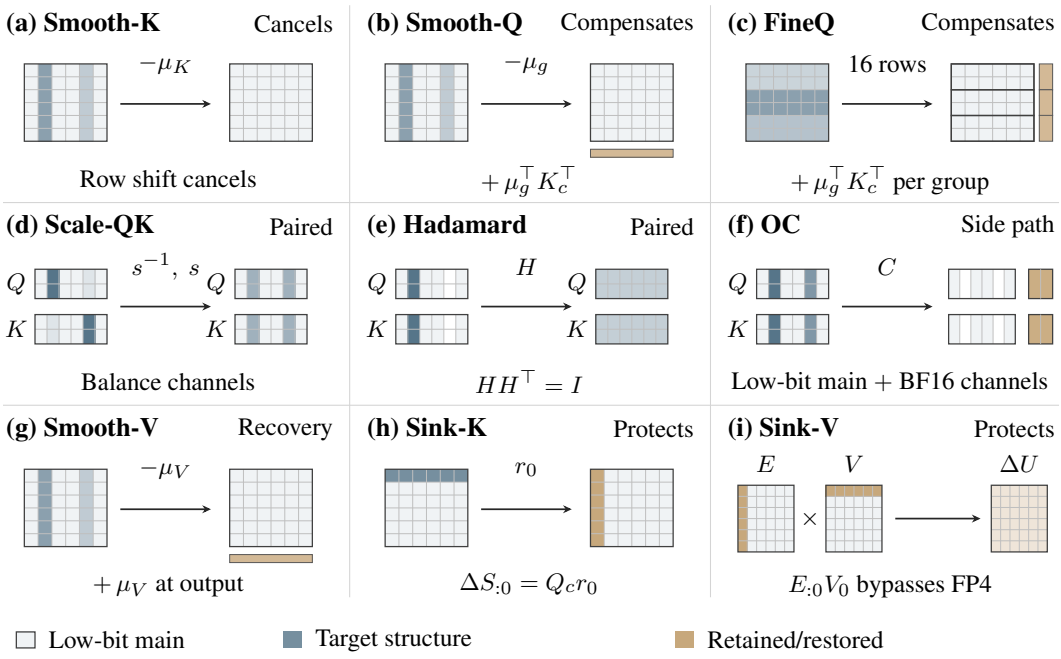


Figure 3: Correction sites and recovery paths. Rows are tokens and columns channels, except S/E columns are keys. Local protection targets key 0. Smooth-Q/FineQ use centered keys; Smooth-V shows the unquantized recovery identity.

Cancellation and shared recovery. Let $K_c = K - \mathbf{1}\mu_K^\top$, $V_c = V - \mathbf{1}\mu_V^\top$, and A be invertible. In exact arithmetic, paired transforms $Q' = QA$, $K' = K_c A^{-\top}$ preserve similarities up to a rowwise shift $c = Q\mu_K/\sqrt{D}$. Using V_c in the value product and means shared across all KV blocks,

$$P' = P, \quad o = o' + \mu_V^\top, \quad (m_t', \ell_t', U_t') = (m_t - c, \ell_t, U_t - \ell_t \mu_V^\top). \tag{5}$$

Diagonal A gives inverse Q/K scaling; orthogonal A gives a shared rotation. No extra compensation product is required, although statistics and transforms still cost time. Sharing K/V means across blocks is essential to this online identity.

Explicit score and output contributions. Query centering $Q - G$ requires adding $GK_c^\top/\sqrt{D}$ before softmax. Basic shares each query mean over a tile; FineQ uses 16-query groups and incurs finer-grained compensation. Retained channels similarly contribute $Q_C K_C^\top/\sqrt{D}$ through ΔS.

Sink-K estimates a selected key's residual and repairs its score column; Sink-V evaluates $E_{:0}v_0^\top$ in higher precision and adds it through ΔU, with the same online rescaling as the main accumulator. Protecting PV does not remove QK error already present in its weight.

Quantized weights change recovery. For effective weights $\widehat{P}$ after encoding and normalization,

$$\widehat{P}(V - \mathbf{1}\mu_V^\top) + \mathbf{1}\mu_V^\top = \widehat{P}V + (\mathbf{1} - \widehat{P}\mathbf{1})\mu_V^\top. \tag{6}$$

Thus direct value-mean restoration can leave a residual when quantization changes row mass. Composition must match mean groups, denominator choice, transform coordinates, and recovery order: FineQ replaces Basic's groups, and rotation with channel protection requires calibration in the rotated coordinates. Appendix B.1 specifies the supported combinations and derivations.

3.5 Executing the Complete Attention Call

FreeAttn reduces repeated work at three points in Figure 2. **Input preparation** shares statistics among centering, compensation, and recovery, and fuses transforms with quantization and packing. Supported short inputs combine Q/K/V means; channel protection prepares its low-precision and BF16 components together. **Online encoding** uses native F2FP for each new weight block, keeping the encoded bytes compatible with PV. **Output recovery** directly writes the requested layout or combines crop, layout conversion, and mean restoration; causal compensation is generated only for tiles that can be read.

The optimized D128 model path shares each Q scale over 32 tokens and each K scale over 64 tokens, across all channels. Their product is folded into score normalization. A fixed exponent offset computes online weights as $16\exp(s - m)$, which native conversion consumes directly. The unencoded denominator carries the same factor, so final normalization cancels it. Per-channel V scales are restored after normalized PV. This separates scale handling from matrix execution and avoids an additional weight-scaling operation before encoding.

FreeAttn selects work assignment according to batch and KV working-set size, retaining causal load balance while improving reuse within groups of query heads. These choices can change grouping or numerical order, so Section 4.1 separates optimized performance configurations from matched-format fidelity controls. Appendix D.1 specifies the model execution policy; Appendix D.4 gives paired ablations and interface boundaries.

4 Evaluation

4.1 Setup and Comparison Scope

The main operator and model benchmarks run on individual GPUs in a server equipped with eight NVIDIA RTX PRO 6000 Blackwell Server Edition GPUs, each with 96 GB of memory. Workloads comprise captured Qwen/Wan attention inputs, Qwen3-8B/32B language models, and Wan2.1-T2V-14B video generation. The compared implementations are listed in Table 1; software versions and workload settings are in Appendix D.

Table 1: Arithmetic configurations of the evaluated attention implementations.

Implementation	QK operands	PV operands
PyTorch SDPA	BF16	BF16
FlashAttention-4 (FlashAttention contributors, 2026)	BF16	BF16
cuDNN attention (NVIDIA, 2026a)	BF16	BF16
SageAttention2++ (Zhang et al., 2025b)	INT8	FP8
SageAttention3	FP4	FP4
FreeAttn	FP4 / FP8 E3 or E4	FP4 / FP8 E3 or E4

Sage2++ uses per-warp quantization; its FP32+FP16 and FP32+FP32 accumulation modes are reported separately. Sage3 uses the official FP4 path. Required K/V replication, input preservation, and layout conversion are included in complete calls. Raw disables corrections; Basic centers Q/K, and FineQ refines the query groups.

Operator error uses FP32 attention with TF32 disabled; model fidelity uses BF16 Torch SDPA. Kernel time excludes preparation, while complete-call time includes fresh preparation and output processing. Qwen replaces prefill attention, retaining BF16 decoding and KV caches; Wan replaces self-attention, retaining BF16 text encoding, cross-attention, and VAE. First-token and MP4-delivery timings include these unchanged components, so they measure the attention savings realized by each model pipeline. All external speedups use the corresponding Torch timing. The primary model and operator results use the same optimized all-E3 configuration. Independent matched E3/E4 controls isolate representation effects; they are not combined with current timing to claim equal-fidelity speedup (Appendix D.1).

4.2 Representation and Asymmetric Precision

At a fixed eight-bit width, E3M4 provides a useful alternative to E4M3. On 18 held-out Qwen states, changing the formats in a controlled FP32-computation experiment reduces aggregate output RMSE by 47.7%. Matched native E3/E4 executions preserve this ordering on all nine captured Qwen/Wan layers (Appendix Figure 8).

Precision allocation also depends on the input. QK8PV4 has lower error than QK4PV8 on the three model captures in Appendix Figure 11, while all six synthetic inputs reverse this ordering. The error decomposition explains why improving PV alone can have limited effect: on six Wan captures, replacing E4 by E3 in isolated PV gives an RMSE ratio of 0.499, but the ratio becomes 0.997 with Raw QK4. Propagated QK error then dominates the output. The corresponding interventions and value-dependent counterexamples are detailed in Appendix B.

The value dependence is visible even when probability error improves: on one Wan1.3B capture, E3 more than halves attention KL yet increases output RMSE. The altered weights act on different value directions, so probability similarity alone does not determine the better precision allocation (Appendix B.3).

4.3 Kernel, Complete Call, and Execution Mechanisms

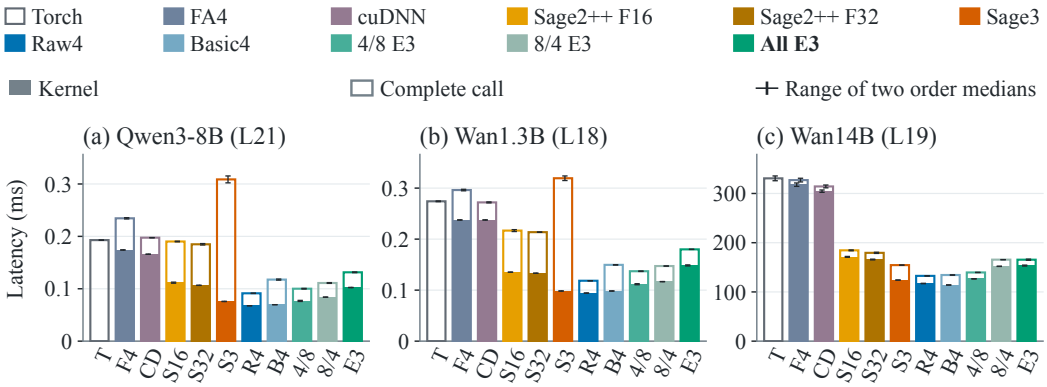


Figure 4: Kernel and complete-call latency on model captures. Filled bars show the kernel; outlines show the complete call. Ranges span two timing orders.

Figure 4 reports kernel and complete-call latency on identical inputs; Appendix Figure 11 pairs these costs with output error. Full E3 achieves 1.47–$2.00\times$ complete-call speedup over Torch while maintaining lower operator error than the FP4 and asymmetric FreeAttn paths on these captures. Basic FP4 reduces Raw FP4 error and remains faster than Sage3 in all three complete-call comparisons. The kernel and complete-call measurements in Appendix D.1 expose the effect of preparation: on Wan1.3B, Sage2++ has a faster attention kernel, whereas E3 has a faster complete call. The synthetic panel extends the comparison to multiple batches and $D = 64/128/256$.

Native conversion at fixed arithmetic. The native/software E3 control preserves outputs bitwise and retains the same register and shared-memory requirements. Native F2FP reduces complete-call time by 4.6–17.4% across the three head dimensions (Figure 5a). Shared statistics, causal-compensation pruning, and output fusion provide additional savings; Table 7 isolates each optimization against its own control.

4.4 When Correction Pays Off

Corrections address different error structures and incur different costs. Refining query-mean groups from 128 to 16 reduces RMSE by 44.5% and increases complete-call time by 12.3% in the single-factor Wan14B control. Lower operator error need not improve model fidelity: fixed Smooth-K reduces error on all three Server captures, but increases mean Qwen3-8B KL from 0.02201 to 0.02772 on the same 32 requests (Appendix E). The earlier video control likewise shows lower layer RMSE but worse BF16 agreement for FineQ/OC (Figure 5b,c). The menu and composition results are retained in Appendix C and Figure 13.

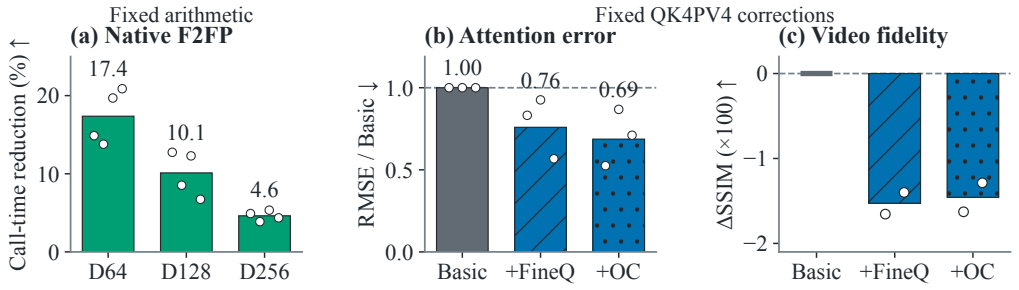


Figure 5: Mechanism controls: (a) native/software E3; (b) correction RMSE on three layers; (c) video SSIM change on one prompt/seed in two orders. Each correction uses its matched Basic.

4.5 Model Fidelity and End-to-End Benefits

Language. Figure 6 evaluates the current E3 policy on identical long inputs for quality and cost. Across 32 requests per model (Bai et al., 2024), Qwen3-8B/32B TTFT decreases from 1.648/6.161 s with Torch to 1.415/5.345 s. E3 is slightly faster than Sage2++ F32 (1.448/5.437 s), while its mean KL to BF16 is higher on both models; the timing advantage over F32 remains below 3%. On the 5,120-question panel with inputs up to 2K tokens, E3-minus-BF16 macro-accuracy differences are $-0.098/-0.059$ percentage points, with paired intervals spanning zero (Table 9). The separate matched E3/E4 controls retain the representation benefit (Table 10).

Video. On four fixed Wan2.1-T2V-14B scene/noise pairs, the same optimized E3 policy reduces geometric-mean delivery time from 2,233 to 1,557 s ($1.43\times$ over Torch), versus 1,616 s for Sage2++ F32. BF16-relative SSIM is 0.692 for E3 and 0.810 for F32. This measures reference agreement, not perceptual quality: the menu examples include an output with higher SSIM but a duplicated object (Appendix F). An independent matched-format study retains an E3/E4 pixel-MSE geometric ratio of 0.587 across six pairs; its numerical policy and timing remain separate (Appendix F).

5 Related Work

FlashAttention combines tiling and online softmax; subsequent versions improve work partitioning, overlap, and Blackwell execution (Dao et al., 2022; Dao, 2024; Shah et al., 2024; Zadouri et al., 2026). INT-FlashAttention and SageAttention use integer QK; SageAttention2 refines granularity, SageAttention2++ accelerates FP8 accumulation, and SageAttention3 adopts microscaled FP4 (Chen et al., 2024; Zhang et al., 2024b;a; 2025b;a). FreeAttn adds a native E3M4 conversion-to-MMA path and studies its output-level numerical effects.

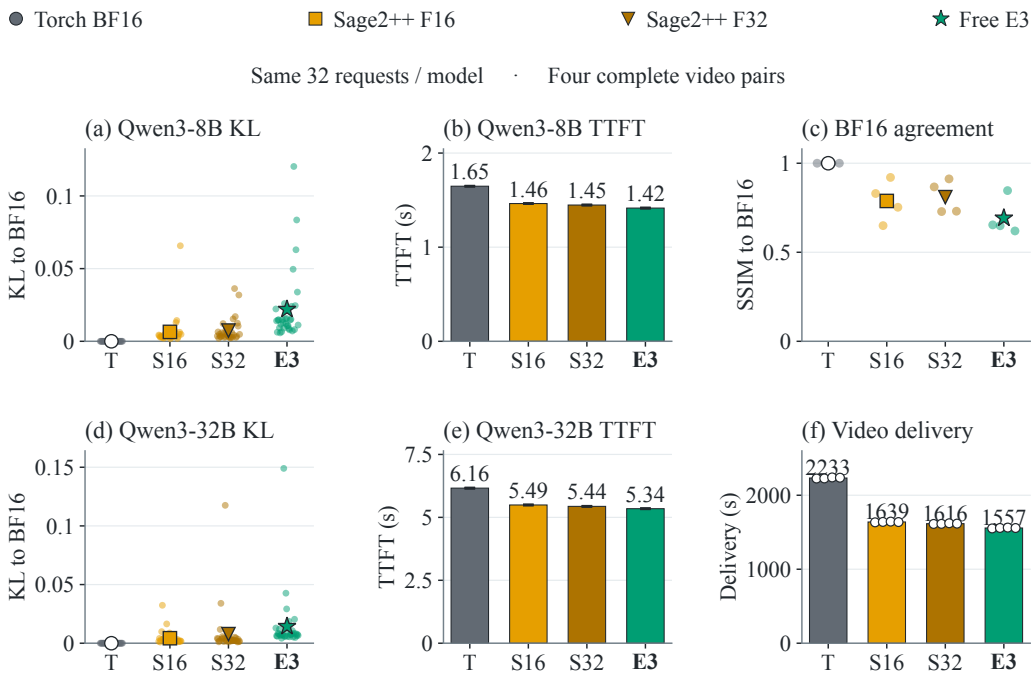


Figure 6: Same-configuration model fidelity and cost. Qwen panels pair KL and TTFT on the same 32 requests; video panels use four scene/noise pairs. Points are requests or complete videos; large markers/bars show means/geometric means. TTFT ranges span the two timing orders.

Prior FP8 analysis relates exponent allocation to operand distributions (Kuzmin et al., 2022). Shared microexponents and MX formats separate group scale from element representation (Darvish Rouhani et al., 2023a;b; Micikevicius et al., 2022); AdaMX and MX-SAFE explore adaptive representations (Luo et al., 2026; Park et al., 2026). Blackwell ISA analyses and CUDA Tile IR also describe E3M4 (kacper-daftcode, 2026; NVIDIA, 2026c). FREEATTN independently validates conversion and matrix interpretation, then integrates both into online attention. Integrating a format into attention additionally requires efficient encoding of the weights generated inside the softmax loop. The fixed-arithmetic native/software comparison isolates this conversion cost from matrix throughput and representation accuracy.

SmoothQuant, Outlier Suppression+, QuaRot, and SpinQuant reshape operand distributions, while LLM.int8() and SVDQuant retain selected contributions in higher precision (Xiao et al., 2023; Wei et al., 2023; Ashkboos et al., 2024; Liu et al., 2025; Dettmers et al., 2022; Li et al., 2025). ThriftAttention mixes block precisions and Attn-QAT adapts training to low-bit attention (Sharratt, 2026; Zhang et al., 2026). FREEATTN instead specifies inference-time composition, compensation, and recovery around softmax without retraining weights.

6 CONCLUSION

FREEATTN connects independently validated E3M4 conversion and MMA into fused attention, combining finer numerical representation with composable correction and efficient execution. Matched-format experiments establish aggregate BF16-relative fidelity gains, while optimized all-FP8 E3 accelerates long-context prefill and video delivery. The analysis and controls show why local quantization error alone cannot determine precision allocation or correction choices. Future work will investigate native E3M4 attention training, focusing on gradient fidelity, convergence stability, and training efficiency.

AI USE STATEMENT

AI tools assisted literature search, research design and analysis, implementation, translation, and language editing; the authors take responsibility for the final content.

REPRODUCIBILITY STATEMENT

Appendices A–G provide instruction probes, numerical derivations, supplementary validation, timing and accuracy contracts, negative controls, and an entry point to the Reproducibility Materials.

REFERENCES

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-free 4-bit inference in rotated LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*, 2024. URL https://arxiv.org/abs/2412.15204.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*, 2020. URL https://arxiv.org/abs/1911.11641.

Shimao Chen, Zirui Liu, Zhiying Wu, Ce Zheng, Peizhuang Cong, Zihan Jiang, Yuhan Wu, Lei Su, and Tong Yang. INT-FlashAttention: Enabling flash attention for INT8 quantization. arXiv preprint arXiv:2409.16997, 2024.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Bita Darvish Rouhani, Ritchie Zhao, Venmugil Elango, Rasoul Shafipour, Mathew Hall, Maral Mesmakhosroshahi, Ankit More, Levi Melnick, Maximilian Golub, Girish Varatkar, Lai Shao, Gaurav Kolhe, Dimitry Melts, Jasmine Klar, Renee L'Heureux, Matt Perry, Doug Burger, Eric Chung, Zhaoxia (Summer) Deng, Sam Naghshineh, Jongsoo Park, and Maxim Naumov. With shared microexponents, a little shifting goes a long way. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, 2023a. URL https://www.microsoft.com/en-us/research/publication/with-shared-microexponents-a-little-shifting-goes-a-long-way/.

Bita Darvish Rouhani, Ritchie Zhao, Ankit More, Mathew Hall, Alireza Khodamoradi, Summer Deng, Dhruv Choudhary, Marius Cornea, Eric Dellinger, Kristof Denolf, Stosic Dusan, Venmugil Elango, Maximilian Golub, Alexander Heinecke, Phil James-Roxby, Dharmesh Jani, Gaurav Kolhe, Martin Langhammer, Ada Li, Levi Melnick, Maral Mesmakhosroshahi, Andres Rodriguez, Michael Schulte, Rasoul Shafipour, Lei Shao, Michael Siu, Pradeep Dubey, Paulius Micikevicius, Maxim Naumov, Colin Verrilli, Ralph Wittig, Doug Burger, and Eric Chung. Microscaling data formats for deep learning. arXiv preprint arXiv:2310.10537, 2023b.

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

FlashAttention contributors. FlashAttention-4 implementation, 2026. URL https://github.com/Dao-AILab/flash-attention. Benchmark version 4.0.0b29.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2009.03300.

Aaron Jarmusch, Nathan Graddon, and Sunita Chandrasekaran. Dissecting the NVIDIA Blackwell architecture with microbenchmarks. arXiv preprint arXiv:2507.10789, 2025.

Zhe Jia, Marco Maggioni, Benjamin Staiger, and Daniele P. Scarpazza. Dissecting the NVIDIA Volta GPU architecture via microbenchmarking. arXiv preprint arXiv:1804.06826, 2018.

kacper-daftcode. Blackwell ISA: Public research artifacts for the NVIDIA Blackwell SASS instruction set, 2026. URL https://github.com/kacper-daftcode/blackwell-isa. Revision 8fe1478; accessed September 13, 2026.

Andrey Kuzmin, Mart van Baalen, Yuwei Ren, Markus Nagel, Jorn Peters, and Tijmen Blankevoort. FP8 quantization: The power of the exponent. *arXiv preprint arXiv:2208.09225*, 2022. URL https://arxiv.org/abs/2208.09225.

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

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. SpinQuant: LLM quantization with learned rotations. In *International Conference on Learning Representations (ICLR)*, 2025.

Junyi Luo, Xinting Jiang, Tai-Hao Wen, Ruichen Qi, Minxing Chu, Hongyi Wu, Gregory Kielian, Ben Laurie, Qirui Zhang, Quan Cheng, Dennis Sylvester, and Mehdi Saligane. Heterogeneity-aware microscaling for efficient low-bit LLM inference. arXiv preprint arXiv:2608.03867, 2026.

Paulius Micikevicius, Dusan Stosic, Neil Burgess, Marius Cornea, Pradeep Dubey, Richard Grisenthwaite, Sangwon Ha, Alexander Heinecke, Patrick Judd, John Kamalu, Naveen Mellempudi, Stuart Oberman, Mohammad Shoeybi, Michael Siu, and Hao Wu. FP8 formats for deep learning. arXiv preprint arXiv:2209.05433, 2022.

Maxim Milakov and Natalia Gimelshein. Online normalizer calculation for softmax. arXiv preprint arXiv:1805.02867, 2018.

NVIDIA. NVIDIA cuDNN: Attention, 2026a. URL https://docs.nvidia.com/deeplearning/cudnn/latest/operations/Attention.html. SDPA frontend documentation; accessed September 13, 2026.

NVIDIA. Parallel Thread Execution ISA, version 9.4, 2026b. URL https://docs.nvidia.com/cuda/parallel-thread-execution/index.html. Accessed September 13, 2026.

NVIDIA. CUDA Tile IR: Types, 2026c. URL https://docs.nvidia.com/cuda/archive/13.2.0/tile-ir/13.1/sections/types.html. Archived CUDA 13.2 documentation; accessed September 13, 2026.

Dahoon Park, Jahyun Koo, Sangwoo Hwang, and Jaeha Kung. MX-SAFE: Versatile inference- and training-proof microscaling format with on-the-fly exponent and mantissa bit allocation. arXiv preprint arXiv:2605.24391, 2026.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. FlashAttention-3: Fast and accurate attention with asynchrony and low-precision. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Joe Sharratt. ThriftAttention: Selective mixed precision for long-context FP4 attention. arXiv preprint arXiv:2605.23081, 2026.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. Massive activations in large language models. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=F7aAhfitX6.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. URL https://papers.nips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Wan Team. Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314, 2025.

Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. Outlier Suppression+: Accurate quantization of large language models by equivalent and effective shifting and scaling. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1648–1665, 2023. doi: 10.18653/v1/2023.emnlp-main.102. URL https://aclanthology.org/2023.emnlp-main.102/.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning (ICML)*, 2023.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.17453.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025.

Ted Zadouri, Markus Hoehnerbach, Jay Shah, Timmy Liu, Vijay Thakkar, and Tri Dao. FlashAttention-4: Algorithm and kernel pipelining co-design for asymmetric hardware scaling. arXiv preprint arXiv:2603.05451, 2026.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019. URL https://arxiv.org/abs/1905.07830.

Jintao Zhang, Haofeng Huang, Pengle Zhang, Jia Wei, Jun Zhu, and Jianfei Chen. SageAttention2: Efficient attention with thorough outlier smoothing and per-thread INT4 quantization. arXiv preprint arXiv:2411.10958, 2024a.

Jintao Zhang, Jia Wei, Haofeng Huang, Pengle Zhang, Jun Zhu, and Jianfei Chen. SageAttention: Accurate 8-bit attention for plug-and-play inference acceleration. arXiv preprint arXiv:2410.02367, 2024b.

Jintao Zhang, Jia Wei, Pengle Zhang, Xiaoming Xu, Haofeng Huang, Haoxu Wang, Kai Jiang, Jianfei Chen, and Jun Zhu. SageAttention3: Microscaling FP4 attention for inference and an exploration of 8-bit training. arXiv preprint arXiv:2505.11594, 2025a.

Jintao Zhang, Xiaoming Xu, Jia Wei, Haofeng Huang, Pengle Zhang, Chendong Xiang, Jun Zhu, and Jianfei Chen. SageAttention2++: A more efficient implementation of SageAttention2. *arXiv preprint arXiv:2505.21136*, 2025b. URL https://arxiv.org/abs/2505.21136.

Peiyuan Zhang, Matthew Noto, Wenxuan Tan, Chengquan Jiang, Will Lin, Wei Zhou, and Hao Zhang. Attn-QAT: 4-bit attention with quantization-aware training. arXiv preprint arXiv:2603.00040, 2026.

Appendices A–C detail native instructions, numerical rules, and correction derivations. Appendices D–F provide operator, language-model, and video protocols and supplementary results. Appendix G describes the reproduction package and archived evidence.

A Semantic Validation of Native Instructions

Section 3.2 and Figure 2 describe format-bit rewriting. This appendix provides independent numerical checks and architectural scope; Table 2 summarizes coverage.

Table 2: Native E3 semantic coverage. Generation and consumption are checked against independent references. SM90a is a format control. QMMA enumerates four format combinations, both operands, and all 256 encodings per operand.

Platform	Primitive	Coverage	Result
SM120	`F2FP`	128 packed pairs per format	Reference match
	`QMMA`	$4 \times 2 \times 256$ bytes + 20 scale checks	Reference match
SM110a (Thor)	`F2FP`	254 finite encodings	Reference match
	Descriptor MMA	4×256 encoding cases	Reference match
SM90a (H200)	`QGMMA`	16 field settings	E4/E5 only

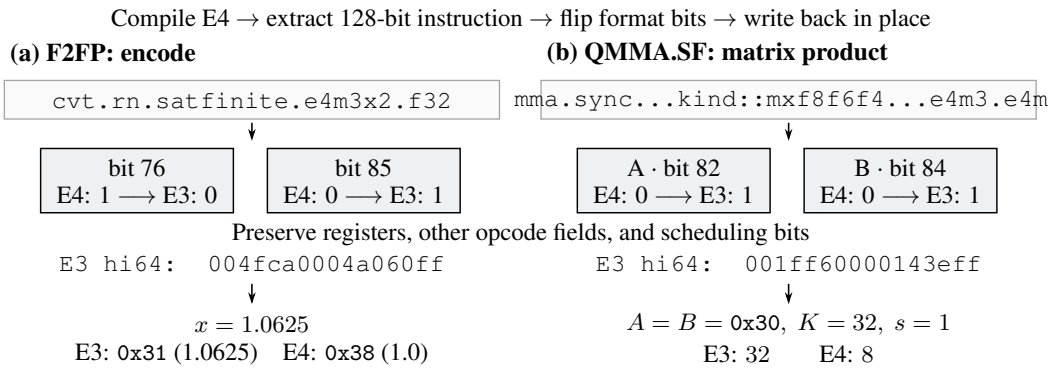


Figure 7: Detailed format-bit recovery and independent conversion/MMA examples. Only the specified selector bits change in each compiled 128-bit instruction.

Extracting instruction words. The rewriter combines the ELF code-section file offset with the disassembly address to locate each 16-byte word, then reinserts the same-length cubin. For `QMMA.SF.16832`, differential fields relative to E2M1 are $\{78, 82, 83\}$ for A and $\{79, 84, 85\}$ for B; modes 5 and 7 identify E4M3 and E3M4 in this differential encoding. They are not PTX type identifiers. Complete binary differences confine edits to the admitted format selectors.

A.1 SM120 Encoding and Matrix Semantics

For finite encodings, an independent E3M4 decoder uses sign s, exponent e, and integer significand field m:

$$d(s, e, m) = (-1)^s \begin{cases} (m/16)2^{-2}, & e = 0, \\ (1 + m/16)2^{e-3}, & e > 0. \end{cases} \tag{7}$$

All-one exponent and significand fields denote NaN. The smallest positive subnormal is $1/64$, the largest finite magnitude is 30, and signed zero is retained. On SM120, differential modes 2/3 relative to E2M1 cause execution faults; these numbers are unrelated to Thor descriptor types.

Matrix consumption. Each of the four E3/E4 combinations enumerates all 256 A and B encodings independently, with the other operand fixed to numerical 1. This prevents paired encoding errors from canceling in the product. Scale controls and expected NaNs are checked separately; finite references explicitly reject NaN/Inf. All 2,068 cases match the independent decoder, and memcheck reports no errors. The native verification records retain case coverage and expected-NaN lists.

Encoding generation. Boundary inputs check packed-slot order, finite saturation, and round-to-nearest ties-to-even. Tests include the packed pairs in Table 2 and extended nonnegative P and signed Q/K/V inputs; they do not enumerate all FP32 bit patterns. Integration checks across three head dimensions and three FP8 routes verify that rewriting leaves unrelated instructions and NVFP4 scales unchanged.

A.2 Format Cost under a Fixed Implementation

An all-FP8 comparison with fixed source, layout, schedule, and corrections does not resolve an E3/E4 device-latency difference: the geometric latency ratio is 1.000160, with a paired 95% bootstrap interval of 0.999907–1.000453. Both mixed-route ratios differ from 1 by less than 0.04%. The control uses B1/H4/D128 and $N = 2^7, \ldots, 2^{11}$, full/causal attention, and ten AB/BA pairs. CUDA events measure complete device work.

A.3 Cross-Architecture Scope

SM110a (Thor). Descriptor type 2 of `tcgen05.mma` has E3M4 semantics. Independent references validate both matrix consumption and F2FP generation (Table 2). This evidence covers Thor primitives; complete attention and SM100 execution remain unverified.

SM90a (H200). All 16 settings of bits 76–79 in the tested `QGMMA.64x8x32.F32` instruction match E4M3/E5M2 controls and do not exhibit E3M4 semantics.

B Quantization Rules and Error Analysis

This appendix defines native quantizers and online denominators, then validates error reconstruction and representation-range controls. Numerical interventions explain error sources and are reported separately from native execution timings.

B.1 Quantizers and Online Denominators

The following quantizers define the matched-format and correction controls. The reference synthetic suite uses the menu-tensor implementation; numerical controls, menu experiments, and matched-format Wan14B evaluations use the workspace-mask implementation, with the denominator rules specified below. The optimized Server configurations in Appendix D.1 use their separately stated scaling policies.

FP8. Each Q/K group spans 32 consecutive channels of one token; each V group spans 32 tokens of one channel. Online weights use the corresponding K32 contraction groups. Let R_f encode and decode with round-to-nearest ties-to-even and finite saturation, with $F_f = 30$ for E3M4 and $F_f = 448$ for E4M3. For a nonzero Q/K/V group, set $a = \max_{i \in b} |x_i|$ and compute exponent e using Equation 3. If the floating-point boundary check gives $a > F_f 2^e$, increment e. Native exponents are clamped to $[-126, 127]$ and stored as UE8M0 bytes $e + 127$. All-zero groups store zero data, scale byte 0, and inverse scale 0. Saturation still applies at representable boundaries.

For online weights $E = \exp(S - m)$, native execution derives scales directly from score maxima: $e = \mathrm{clip}(\lceil (\max_b S - m)/\ln 2 - \log_2 F_f \rceil, -126, 127)$. Fully masked groups produce zero with the minimum exponent. Some raw mixed paths compute exponentials directly in the scaled domain, changing floating-point operation order. FP32 numerical controls instead clamp exponents to $[-127, 127]$ and use scale 1 for all-zero groups. Native reconstruction checks remain necessary: the control is not a bitwise simulation at every exponent boundary.

FP4. Let R_4 be a round trip through a nonnegative E4M3 scale and R_2 an E2M1 conversion, both using nearest-even rounding and finite saturation. The main runtime's E4M3/block-16 path uses

$$a_{QK} = R_4\left(\max\left(\max_b |X|/6, 10^{-7}\right)\right),$$
$$a_V = R_4\left(\max\left(\max_b |V_c|/6, 2^{-9}\right)\right), \tag{8}$$
$$Z = 2688E, \qquad a_P = R_4\left(\max\left(448 \max_b E, 5.25\right)\right).$$

Here X is prepared BF16 Q/K and V_c is BF16 V, optionally centered. Q/K uses a BF16-rounded reciprocal and BF16 multiplication before R_2. V uses an approximate FP32 reciprocal rounded to BF16 before BF16 multiplication. P converts from FP32 Z/a_P. These rounding steps are part of the operator. The Q/K floor 10^{-7} can itself round to zero in E4M3, in which case the stored group reconstructs to zero. No additional positive floor is imposed, and no universal payload bit pattern is assumed for zero-scale groups. The V and P rules yield positive stored scales.

Q/K block-16 groups use contiguous channels. P/V groups follow MMA packing: within a 32-token region, $G_0 = \{8j + k : 0 \le j < 4,\ 0 \le k < 4\}$ and $G_1 = G_0 + 4$. Replacing them with two contiguous 16-token halves defines a different quantizer. FP8's complete K32 groups are unaffected by this permutation.

Online denominator and effective weights. The common P4 factor 2688 scales both numerator and denominator and cancels during normalization. The recurrence below omits it, expressing decoded weights $\widehat{E}$ in units of E. Ordinary paths use $D_t = E_t$; the long-sequence tail NVFP4 path in workspace-mask uses $D_t = \widehat{E}_t$. Let m_t be the running maximum, $\alpha_t = e^{m_{t-1}-m_t}$ the rescaling factor, and T the last block. Before output rounding, an unprotected key j in block t has effective weight

$$\widehat{p}_j = \frac{e^{m_t - m_T}\,\widehat{E}_{t,j}}{\ell_T}, \qquad \ell_t = \alpha_t \ell_{t-1} + \mathrm{rowsum}(D_t). \tag{9}$$

Using encoded weights in the denominator normalizes their reconstructed row sum in real arithmetic, but V quantization and recovery rounding remain. Sink-V's protected column uses saved unencoded weights.

The encoded-weight branch requires D128 full self-attention, logical $N \ge 8192$, a logical tail, and Basic, Basic+Smooth-V, Basic+Hadamard, or their three-way combination. It uses M128/N128 and is absent from the synthetic cohort. Table 3 lists the selected paths for Wan14B at $N = 75{,}600$, padded to 75,648.

Table 3: Wan14B QK4PV4 option semantics. The compute query tile is $M = 128$; FineQ changes mean groups to 16. Public-option comparisons include these schedule and denominator differences. H/V denote Hadamard/Smooth-V.

Option	Q mean group	KV tile	Denominator
Raw	–	128	Unencoded E
Smooth-K	–	64	Unencoded E
Basic / Basic+H / +V / +H+V	128	128	Reconstructed $\widehat{E}$
Basic+FineQ	16	64	Unencoded E
Basic+Scale-QK / +OC / +OC+V	128	64	Unencoded E

Logical masks and tails. Raw compact FP8 paths provide zero for out-of-range quantized inputs. Corrected adapters pad incomplete query groups with their BF16 means. Smooth-K/V pads with the corresponding logical-sequence means; other paths use zero. Groups follow the selected coarse or FineQ schedule. Scores beyond logical KV length or in the causal future are set to $-\infty$ and contribute no mass. Outputs are cropped to the logical query length. Mean padding and denominator selection are independent.

Appendix C.3 gives composition conditions. Public configurations in Table 3 and fixed-denominator single-factor controls are reported separately.

B.2 Validating the Error Reconstruction

Error attribution requires a numerical reconstruction that accounts for native outputs; otherwise, quantizer and accumulation-order differences contaminate propagation effects. Qwen3-8B captures six 2,048-token WikiText-2 windows at layers 0/7/14/21/28/35. Three offsets are used for calibration and three for validation, producing 18 states each. Controls use 128 fixed queries and all keys. Heads are correlated observations, and no selector is fitted.

Two native execution orders cover 36 states and four raw/corrected E3/E4 configurations. All 288 reconstructions have RMSE from the native output below both $\max(10^{-5}, 0.005\,\mathrm{RMS}_{\mathrm{ref}})$ and $\max(10^{-5}, 0.1\,\mathrm{RMSE}_{\mathrm{native}})$. Here $\mathrm{RMS}_{\mathrm{ref}}$ is the FP32 reference output RMS, and $\mathrm{RMSE}_{\mathrm{native}}$ is native error against that reference. Reconstruction uses decoded native operands and the actual BF16 score compensation without bitwise simulation of native accumulation. Each E3/E4 pair fixes native execution geometry.

Attribution retains the cross term in Equation 4: $\|e_{\mathrm{QK}} + e_{\mathrm{PV}}\|_F^2 = \|e_{\mathrm{QK}}\|_F^2 + \|e_{\mathrm{PV}}\|_F^2 + 2\langle e_{\mathrm{QK}}, e_{\mathrm{PV}}\rangle_F$. In software controls on the 18 validation states, each state's terms are normalized by total error energy before arithmetic averaging. For raw QK4/PV E4, QK, PV, and cross-term shares are 98.95%, 1.07%, and -0.02%; with Smooth-QK they are 95.26%, 4.48%, and 0.26%. QK error dominates these controls; the cross term is small and does not explain the masked PV improvement. These shares describe the software controls. Native agreement is established by the reconstruction checks above, and component RMSEs are not additive.

B.3 Representable Range and Output Propagation

E3 changes both within-group resolution and small-value coverage. FP32 interventions restore either values rounded to zero by E3 or all values in its subnormal range, testing whether small-value loss explains output differences. Q/K groups span 32 channels; V and streaming-weight groups span 32 keys. Under both interventions, the aggregate E3/E4 output-RMSE ratio remains approximately 0.52 for Qwen and 0.65 for Wan1.3B. Restoring zeros changes error by less than 0.01% at Wan14B layers 0/19/39. These are three captures from one forward pass, not three videos. The interventions are numerical controls, not native kernels or timing results.

Format improvements in isolated PV can be altered by QK error. On six additional Wan1.3B captures, the isolated-PV E3/E4 RMSE ratio is 0.499; it becomes 0.997 with raw QK4 and 0.990 with Smooth-QK. Corrected full-attention E3 in this diagnostic is a software counterfactual, not an exposed native entry point.

For reference probability row p, output $o = pV$, and QK-induced change δp,

$$\delta o_{\mathrm{QK}} = \sum_j \delta p_j (v_j - o) \approx \sum_j p_j \delta s_j (v_j - o). \tag{10}$$

The equality uses $\sum_j \delta p_j = 0$; the approximation linearizes softmax. At Wan1.3B layer 29, E3 reduces attention KL from 0.481 to 0.205 and probability MSE by 23.6%, but increases output RMSE from 0.0372 to 0.0399. Restoring small values does not remove this counterexample: the direction of weight error relative to V also matters.

Encoding can additionally change the weight row sum and introduce gain error. For normalized p_{QK}, encoded weights $\widehat{p}$, and $c = \sum_j \widehat{p}_j$,

$$(\widehat{p} - p_{\mathrm{QK}})V = (c-1)p_{\mathrm{QK}}V + (\widehat{p} - c p_{\mathrm{QK}})V. \tag{11}$$

The first term changes gain; the second has zero-sum weights. Diagnostic renormalization reduces isolated P error but changes aggregate complete-output RMSE by only approximately 0.3%.

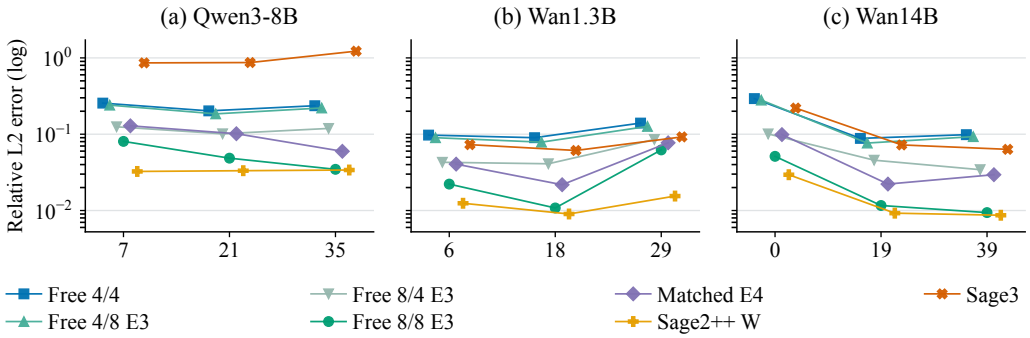


Figure 8: Matched-format comparisons on nine captured layers. Errors use FP32 attention; each point is one fixed layer.

C Algebraic Conditions and Composition Rules

This appendix links captured activation structure to the cancellation, compensation, and composition rules used in the main text.

C.1 Structure in Captured Activations

Figure 9 contrasts channel offsets and local attention concentration in Qwen3-8B. The fixed captures use layers 0/7 of one 2,048-token window, with batch/head 0 and the corresponding KV head. The panels show the first 128 tokens and all 128 channels. Reference P is normalized over all visible keys in FP64 before cropping. Statistics use all tokens of the selected head, and attention mass is averaged over all queries.

Channel scale and shared offsets are distinct structures. In the displayed Qwen L0 head, the largest K-channel norm is $77.5\times$ the median, while the shared mean accounts for 94.0% of K energy, measured as $N_k\|\mu_K\|_2^2/\|K\|_F^2$. These motivate channel balancing/protection and shared-offset removal, respectively. Positional effects depend jointly on weights and values: L7 assigns key 0 an average 39.7% of attention mass, although its V-row norm is only 19.1% of the median of other rows. Its row error is therefore heavily weighted in PV, motivating local protection.

C.2 Derivation of Equivalent Transforms and Explicit Compensation

Consider one head with $Q \in \mathbb{R}^{N_q \times D}$ and $K, V \in \mathbb{R}^{N_k \times D}$, a fixed mask M, and at least one visible key per query. All identities in this subsection use real arithmetic before quantization. Vectors $\mathbf{1}_q, \mathbf{1}_k$ have lengths N_q, N_k. Let $S = QK^\top/\sqrt{D} + M$, $P = \mathrm{softmax}_{\mathrm{row}}(S)$, and $O = PV$.

Paired Q/K transforms. For invertible A, $(QA)(KA^{-\top})^\top = QK^\top$. A diagonal $A = \Lambda^{-1}$ divides Q and multiplies K by the same channel scales, giving Scale-QK. An orthogonal Hadamard matrix $A = H$ satisfies $A^{-\top} = H$, so both operands use the same rotation. The two sides must be transformed together.

Key centering and softmax. Let $K_c = K - \mathbf{1}_k\mu_K^\top$ with a single shared offset, and define $Q' = QA$, $K' = K_cA^{-\top}$. Then

$$Q'K'^\top = Q(K - \mathbf{1}_k\mu_K^\top)^\top = QK^\top - (Q\mu_K)\mathbf{1}_k^\top. \tag{12}$$

Thus $S' = S - c\mathbf{1}_k^\top$, where $c = Q\mu_K/\sqrt{D}$. For query i and visible key j, the same factor cancels from numerator and denominator:

$$p'_{ij} = \frac{e^{s_{ij}-c_i}}{\sum_{k\in\mathcal{V}_i} e^{s_{ik}-c_i}} = \frac{e^{s_{ij}}}{\sum_{k\in\mathcal{V}_i} e^{s_{ik}}} = p_{ij}. \tag{13}$$

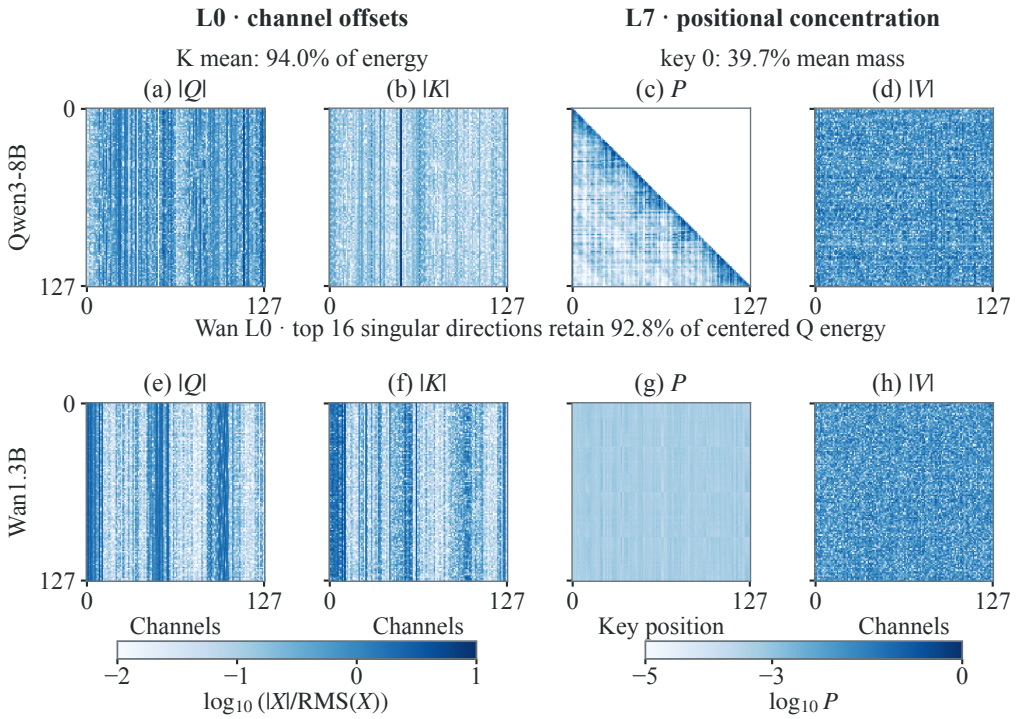


Figure 9: Structure in captured Q/K/V and attention weights. Qwen Q/K use layer 0; P/V use layer 7. Colors show log magnitude relative to head RMS for Q/K/V, and log probability for P. White marks the causal mask.

Masked weights remain zero. Cancellation holds for full and causal attention but requires the same offset across every KV block.

Value centering and recovery. Let $V' = V - \mathbf{1}_k \mu_V^\top$. Since $P' = P$ and $P\mathbf{1}_k = \mathbf{1}_q$,

$$P'V' = P(V - \mathbf{1}_k \mu_V^\top) = O - \mathbf{1}_q \mu_V^\top, \qquad O = P'V' + \mathbf{1}_q \mu_V^\top. \tag{14}$$

Smooth-V therefore restores the output by broadcasting the offset. Combining this with paired Q/K transforms gives Equation 5.

Recovery under online softmax. For query q and KV block t, let $s_t = (K_t q)^\top / \sqrt{D} + M_t$. The recurrence visits blocks with visible keys; fully masked blocks leave the state unchanged. Online softmax maintains (Milakov & Gimelshein, 2018; Dao, 2024)

$$\begin{aligned} m_t &= \max(m_{t-1}, \max s_t), \quad \alpha_t = e^{m_{t-1} - m_t}, \quad E_t = e^{s_t - m_t}, \\ \ell_t &= \alpha_t \ell_{t-1} + E_t \mathbf{1}, \qquad U_t = \alpha_t U_{t-1} + E_t V_t. \end{aligned} \tag{15}$$

The initial state is $(-\infty, 0, 0)$ and the final output is $o = U_T / \ell_T$.

For fixed query i, every block receives the same score shift $c = c_i$. Its running maximum becomes $m'_t = m_t - c$, giving $E'_t = E_t$, $\alpha'_t = \alpha_t$, and $\ell'_t = \ell_t$. Assuming the invariant at step $t-1$ and using $V'_t = V_t - \mathbf{1}\mu_V^\top$,

$$\begin{aligned} U'_t &= \alpha_t (U_{t-1} - \ell_{t-1} \mu_V^\top) + E_t (V_t - \mathbf{1}\mu_V^\top) \\ &= U_t - (\alpha_t \ell_{t-1} + E_t \mathbf{1}) \mu_V^\top = U_t - \ell_t \mu_V^\top. \end{aligned} \tag{16}$$

The zero initial accumulator establishes the base case. Dividing by ℓ_T and restoring $\mu_V^\top$ proves the output identity.

Explicit compensation for query centering. Let $G_i = \mu_{g(i)}^\top$ collect arbitrary query-group means, and define $Q_G = (Q - G)A$ while retaining K', V' above. Then

$$S_G = Q_G K'^\top / \sqrt{D} + M = S - c\mathbf{1}_k^\top - \Delta S, \qquad \Delta S = G K_c^\top / \sqrt{D}. \tag{17}$$

Unlike c, Δs_{ij} generally varies with key j and does not cancel through softmax. Restoring it gives the full Q/K/V identity:

$$\mathrm{softmax}_{\mathrm{row}}(S_G + \Delta S)V' + \mathbf{1}_q \mu_V^\top = O. \tag{18}$$

Smooth-Q/FineQ explicitly computes this grouped score term. Replacing K_c with original K adds only a rowwise constant and does not change softmax. Query-mean compensation therefore requires additional score computation. After quantization, $\widehat{P}\mathbf{1}_k$ need not equal $\mathbf{1}_q$; Equation 6 and the actual denominator rule determine the remaining recovery error.

C.3 Grouping, Precision, and Compatibility

The identities above require shared K/V offsets across KV blocks and explicit grouped score compensation for Q centering. Native execution also fixes statistical precision, query groups, and recovery order. Means accumulate in FP32 and are stored in BF16; subtraction rounds to BF16. Coarse query groups follow the executed tile: 128 rows for D128, 64 for D256, and usually 128 for D64. When padded length is at least 4096 and divisible by 256, the D64 NVFP4 causal path uses 256 rows, except for FineQ/OC. FineQ replaces coarse means with 16-query means. The separate corrected D128 QK4PV8 E3/E4 entry uses 128-row groups.

Scaling and rotation. Scale-QK must share scales among query heads that share K. The implementation samples padded internal BF16 inputs before centering. For length L, sample positions are $0, d, 2d, \ldots < L$, where $d = \max(1, \lfloor L/256 \rfloor)$. Let $a_{h,c}^Q, a_{k,c}^K$ be sampled channelwise absolute maxima. All query heads sharing KV head k use

$$s_{k,c} = \sqrt{\frac{\max(\max_{h \in \mathcal{G}(k)} a_{h,c}^Q, 10^{-9})}{\max(a_{k,c}^K, 10^{-9})}}. \tag{19}$$

Channels that are zero on both sides receive $s = 1$, without further clipping. Centered Q is divided by s and centered K multiplied by s, with BF16 rounding before rotation. Hadamard uses the fixed normalized Sylvester matrix $H_{ij} = (-1)^{\mathrm{popcount}(i \& j)}/\sqrt{D}$ in natural channel order, with FP32 butterfly operations and BF16 storage. It uses neither random signs nor learned rotations.

OC restricts compensation to one frozen channel set. It decomposes $QK^\top = Q_{\bar{C}} K_{\bar{C}}^\top + Q_C K_C^\top$ and restores the BF16 side product with the common score scale, extending mixed-precision linear decomposition (Dettmers et al., 2022) to attention scores. Calibration first centers Q in 128-row groups for D64/D128 or 64-row groups for D256, and centers K globally per head. Rotated OC then applies the same fixed Hadamard transform. For calibration state a, channel energies are normalized by each head's total energy, floored at 10^{-30}, then summed across states to form $u_{h,c}^Q, u_{k,c}^K$. Each KV group freezes the same ordered 16 channels for Q and K:

$$C_k = \mathrm{Top16}_c \left[\max \left(\max_{h \in \mathcal{G}(k)} u_{h,c}^Q, u_{k,c}^K \right) \right]. \tag{20}$$

These channels are zeroed before low-precision scale computation. Ties follow the recorded `torch.topk` output; serialized indices identify each evaluated policy without an additional tie-breaking rule.

Channel selection is separate from validation and freezes indices before evaluation. The synthetic panel calibrates per shape. Captured-input experiments use three designated calibration inputs per domain and store separate rotated/unrotated indices. The Wan14B menu uses the first conditional BF16 forward pass from an independent seed across all 40 layers, ranking only complete query groups and excluding tails. The combination study reuses the unrotated indices. Experiment records identify calibration seeds and serialized indices.

Sink terms and value recovery. Sink-K estimates a local residual with an auxiliary quantizer, which must be distinguished from the actual packed K. The corrected D128 QK4PV8 entry forms $x_0 = K_0 - \mu_K$ in FP32 and recomputes block-16 scales β using the Q/K rule in Equation 8. It reconstructs $\widetilde{k}_{0,c} = \beta_b T(x_{0,c}/\beta_b)$, where T rounds to E2M1 and breaks midpoint ties toward larger magnitude. It then forms $r_0 = x_0 - \widetilde{k}_0$ and the FP32 dot $c_i = \langle q_{c,i}, r_0 \rangle$, with $q_{c,i} = \mathrm{RN}_{\mathrm{BF16}}(q_i - \mu_{g(i)})$. The auxiliary procedure omits native K's intermediate BF16 rounding and uses a different midpoint rule; it estimates a residual rather than decoding the packed K. A zero auxiliary scale reconstructs the key group to zero. The BF16 Smooth-Q term computed from original K initializes the FP32 QK accumulator. Only column 0 receives c_i before the common $1/\sqrt{D}$ scaling and softmax.

Sink-V directly protects a selected PV contribution by retaining the first key's unencoded weight and a BF16 value (Sun et al., 2024; Xiao et al., 2024). With Smooth-V, $v_{0c} = \mathrm{RN}_{\mathrm{BF16}}(v_0 - \mu_V)$ matches the centered main V. The first block saves E_0, excludes it from FP4 P scale selection, zeros its packed column, and adds $C_{PV} = E_0 v_{0c}^{\top}$ to the FP32 output accumulator, followed by the same online rescaling. The notation omits P4's common factor 2688. Final recovery is

$$o = \mathrm{RN}_{\mathrm{BF16}}(\mathrm{RN}_{\mathrm{BF16}}(U/\ell) + \mu_V)\,. \tag{21}$$

This causal QK8PV4 path uses the ordinary denominator. Protecting V does not remove QK error in the corresponding weights.

Algebraic compatibility and current entry-point support are distinct. Native paths support Sink-K on causal QK4PV8 with Smooth-Q/K and Sink-V on causal QK8PV4. Causal Smooth-V requires Sink-V. FineQ and Scale-QK require FP4 QK. Current OC entries do not support combination with FineQ, Scale-QK, or Sink-K. FineQ replaces coarse query groups, and Hadamard+OC requires calibration in rotated coordinates. Unsupported combinations are rejected. Frozen selection still incurs online preparation; metadata and execution share tile, group, and padding decisions.

D Operator Comparisons and Execution Ablations

The Server comparison below supplies the main performance figures. The following reference suite and controlled ablations isolate format, conversion, and correction effects.

D.1 Server Operator Comparisons

The primary operator experiments use a single RTX PRO 6000 Blackwell Server Edition GPU in the eight-GPU server, with Torch 2.14 and CUDA 13.0. The E3 and NVFP4 libraries are identified by `86ae0ad4` and `d0a2fca2`; the Tensor frontend is built for the installed Torch ABI. Captures are fixed before measurement: Qwen3-8B layer 21, Wan1.3B layer 18, and Wan14B layer 19, retaining their full sequence lengths. Common BSHD inputs preserve captured values; any conversion to the BF16 attention entry and restoration to the original query dtype are charged to complete calls.

Each method is measured in two opposite execution orders after three warmups, with five synchronized complete-call observations per order. Table 4 reports the mean of the two order medians; figure ranges span those medians. Complete per-shape measurements are retained in the reproduction materials. Core timing isolates one attention kernel with independent output and changed-input checks. It excludes preparation and output handling, so subtracting these independent timings is not a measured stage decomposition.

Sage2++ uses per-warp quantization, K centering, and no V centering. The official FP32+FP16 mode uses V scale maximum 2.25, whereas FP32+FP32 uses 448; both modes are reported without selecting per-input winners. Sage3 retains its official FP4 arithmetic and includes input preservation and GQA replication. FA4 and cuDNN use their BF16 paths. D256 Sage2/Sage3 native paths are unsupported; FA4 D256 exceeds the device shared-memory limit and is retained as a failed native configuration, without fallback substitution.

The optimized FreeAttn policies use shared Q/K scales, a common online-weight encoding range, and per-channel V-scale recovery. These policies are fixed throughout the operator comparison. The matched-format model experiments retain their separately specified quantization groups and

correction settings; operator latency and historical model quality are not combined into an equal-fidelity speedup.

Quantization in the current long-context and video path. For the D128 path used in Figure 6, each Q scale spans 32 tokens and all channels; each K scale spans 64 tokens and all channels. For a nonzero group with maximum magnitude a, the power-of-two scale is $2^{\lceil \log_2(a/30) \rceil}$; the Q/K scale product is applied before softmax. Online weights are encoded as $16\exp(s-m)$, with the same factor in the unencoded normalization sum, so it cancels at output normalization. V uses one power-of-two scale per KV head and channel. Causal attention uses the maximum over all tokens; full attention samples every 16th token and multiplies the sampled maximum by 1.25 before selecting the scale. V scales are restored after normalized PV. Native conversion uses nearest-even rounding with finite saturation. The reported long-Qwen and Wan14B inputs select M192/N64; the source package's `src/sm120/host/e3m4_routes_all_dims.cu` contains the fixed shape-dispatch predicates. These rules apply to the optimized model path, separately from the matched-format controls in Appendix B.

Table 4: Complete-call time (ms) and relative L_2 error on the three model captures. F16/F32 denote the two Sage2++ modes.

Method	Qwen3-8B L21		Wan1.3B L18		Wan14B L19	
	Call	Error	Call	Error	Call	Error
Torch BF16	0.193	0.001719	0.274	0.001679	330.746	0.00211
FA4 BF16	0.234	0.001719	0.296	0.001679	327.193	0.00211
cuDNN BF16	0.198	0.001745	0.272	0.001678	314.308	0.00211
Sage2++ F16	0.190	0.03325	0.217	0.009007	184.478	0.009233
Sage2++ F32	0.185	0.03424	0.214	0.009117	179.398	0.009281
Sage3 FP4	0.309	0.869	0.319	0.06123	154.463	0.07239
Free 4/4 Raw	0.091	0.2022	0.118	0.09001	132.604	0.08818
Free 4/4 Basic	0.118	0.1374	0.150	0.06259	134.352	0.07777
Free 4/8 E3	0.100	0.19	0.137	0.07919	139.561	0.08556
Free 8/4 E3	0.111	0.1013	0.147	0.04098	165.584	0.04566
Free 8/8 E3	0.131	0.04854	0.180	0.0108	165.477	0.0267

Table 5: Precision allocation on the six synthetic inputs of Figure 10. Entries are relative L_2 errors; all FP8 operands use E3.

Input	QK4PV4	QK4PV8	QK8PV4	QK8PV8
(a)	0.09585	0.01418	0.09576	0.01358
(b)	0.09467	0.01343	0.09464	0.01325
(c)	0.1442	0.03938	0.1399	0.02026
(d)	0.1419	0.03872	0.1375	0.01953
(e)	0.14	0.03838	0.1357	0.01959
(f)	0.1415	0.03858	0.1375	0.0195

D.2 Reference 16-Shape Operator Protocol

Operator comparisons fix logical inputs, output layout, and timing boundaries before comparing speed on common support. Device-resident Q/K/V are fixed-seed BF16 normal samples scaled by 0.5, with zero dropout and attention scale $1/\sqrt{D}$. Complete calls return newly allocated contiguous output and include allocation and padding, including the K copy required by Sage3 to preserve input immutability. Prepared-core timing includes required output conversions. Calibration uses seed 20260907; selections and OC indices are frozen before validation seeds 20260908–10.

The reference suite uses RTX PRO 6000 Server Edition, driver 590.48.01, Torch 2.14.0+cu130, FA4 4.0.0b29, cuDNN frontend 1.28.0/runtime 92501, Sage2++ 2.2.0, and Sage3 1.0.0. Its 16 shapes cover $B = 1$–8, MHA/GQA, $D = 64/128/256$, $N = 512$–32768, and a 577-token tail in both layouts and masks. The 19 FreeAttn configurations yield 494 executable cells. Each of two process orders uses five warmups and five complete calls; prepared-core measurements use

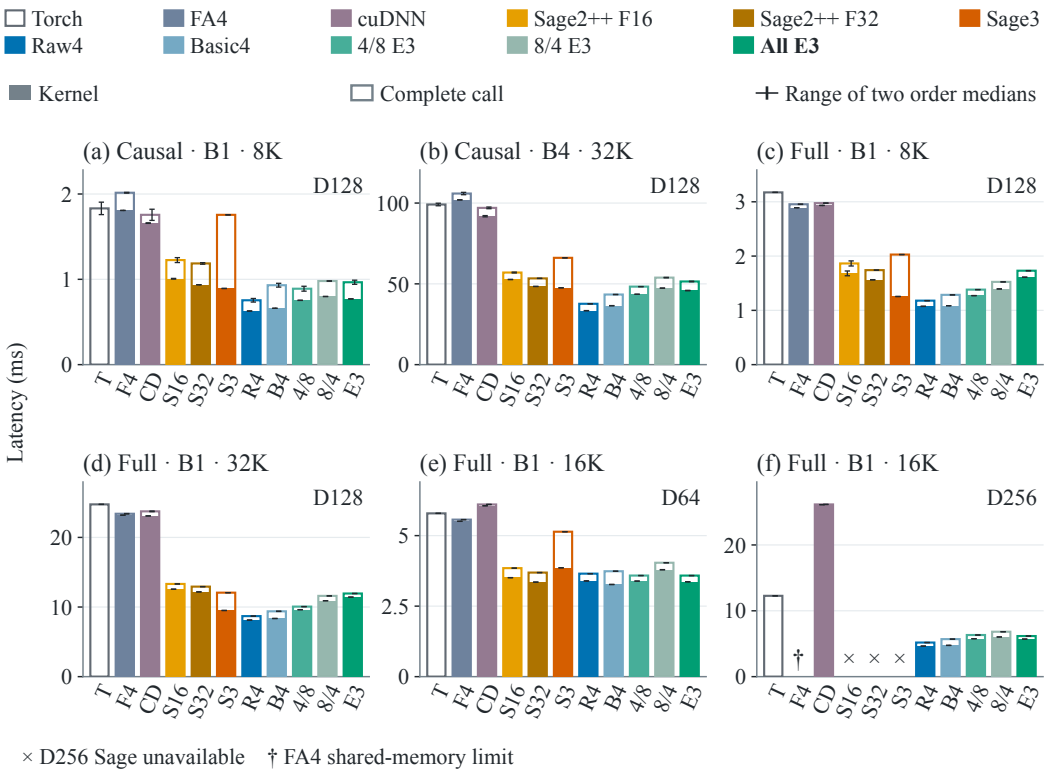


Figure 10: Server synthetic results. Filled bars show kernel time and outlines show complete-call time. Missing native support and failed launches remain marked.

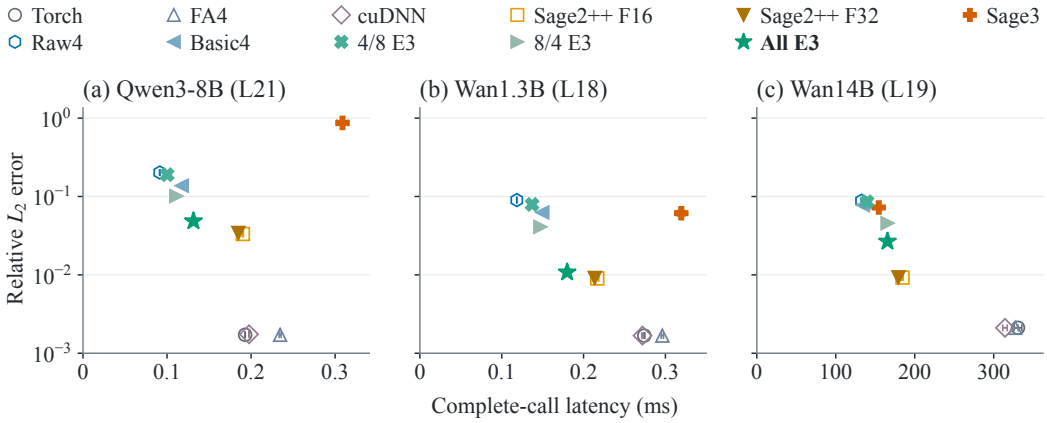


Figure 11: Server error–latency comparison on the same three model captures. Horizontal ranges span two timing orders.

three Graph rounds of at most 32 calls. A resolved timing benefit exceeds 3% in both orders. Strict accuracy requires global cosine ≥ 0.999, fifth-percentile token cosine ≥ 0.99, and RMSE $\leq \max(10^{-5}, 0.05\,\mathrm{RMS}_{\mathrm{ref}})$ on every validation seed and order.

Table 6: Fixed methods on commonly supported shapes. Speedups are paired geometric means relative to Torch. Sage2++ W fixes `fp32+fp16`; the last column counts shapes passing the strict accuracy gate.

SageAttention3: seven common shapes

Method	Core speedup ↑		Call speedup ↑		Strict accuracy passing shapes
	BHSD	BSHD	BHSD	BSHD	
Torch SDPA	1.00	1.00	1.00	1.00	7/7
cuDNN	1.08	1.08	0.99	1.01	7/7
FA4	0.95	1.01	0.77	0.85	7/7
Sage2++ (W)	1.75	1.75	0.99	1.00	7/7
Sage3	2.00	1.80	0.77	0.67	0/7
Free 4/4 Raw	2.34	2.34	1.90	1.92	0/7
Free 4/4 Basic	2.19	2.18	1.51	1.45	0/7
Free 4/8 E3	2.02	2.02	1.65	1.70	7/7
Free 8/4 E3	1.93	1.76	1.53	1.38	0/7
Free 8/8 E3	1.54	1.42	1.28	1.20	7/7

SageAttention2++: twelve common shapes

Method	Core speedup ↑		Call speedup ↑		Strict accuracy passing shapes
	BHSD	BSHD	BHSD	BSHD	
Torch SDPA	1.00	1.00	1.00	1.00	12/12
cuDNN	0.94	0.94	0.95	0.98	12/12
FA4	0.83	0.89	0.70	0.81	12/12
Sage2++ (W)	1.46	1.46	0.87	0.90	12/12
Free 4/4 Raw	2.11	2.11	1.90	1.99	0/12
Free 4/4 Basic	2.00	1.99	1.51	1.49	0/12
Free 4/8 E3	1.86	1.85	1.65	1.69	12/12
Free 8/4 E3	1.73	1.57	1.50	1.38	0/12
Free 8/8 E3	1.37	1.26	1.27	1.20	12/12

Output accuracy is checked before selecting speed-qualified routes. The FP32 reference disables TF32 and uses all queries when $N_q \leq 512$; otherwise it uses 64 uniform queries plus fixed boundary/tail queries, always covering every key. FP64 validates small cases. Table 6 specifies pairing and aggregation; execution orders and layouts do not increase the quality sample count. An accuracy-qualified per-shape selection and a fixed-method comparison are different results. The qualified set over 16 shapes selects core and complete-call routes separately: all 32 complete-call cells select QK4PV8–E3, while three D256 core cells select QK8PV8–E3. Table 6 instead fixes each implementation and the Sage2++ W accumulation mode, normalizing to Torch on each common-support set. The twelve- and seven-shape cohorts are aggregated separately.

The full inventory retains eight FA4 D256 shared-memory failures, eight unsupported Sage2 D256 cells, and eighteen unsupported Sage3 GQA/D256 direct-interface cells. These are separate from the Qwen adaptation that explicitly repeats K/V. Basic FP4/Sage3 RMSE ratios are 0.9845 for full attention and 0.3551 for causal attention; local cosine regressions remain visible despite aggregate gains.

D.3 Captured Layers and Baseline Configurations

Figure 8 compares fixed E3/E4 settings on three Qwen3-8B, three Wan1.3B, and three Wan14B layers. Qwen/Wan1.3B retain BF16 BHSD captures; Wan14B retains 75,600 tokens, 40 heads, FP32 Q/K, and BF16 V. Error uses fixed query positions and all keys. The optimized Server latency comparison is specified separately in Appendix D.1.

Sage2++ configuration controls. Quantization granularity and smoothing can change accuracy and cost, so they are fixed explicitly. Sage2++ W selects per-warp quantization; T uses the installed API's default per-thread granularity. Both use INT8 QK, FP8 PV, `fp32+fp16` accumulation, K centering, and no V centering. Figure 12 compares these granularities and adds a per-warp control with K centering disabled, separating granularity from smoothing. Operator rankings on these fixed captures are not extrapolated to model outputs. Probability, task, and TTFT evaluations use the explicitly selected W configuration.

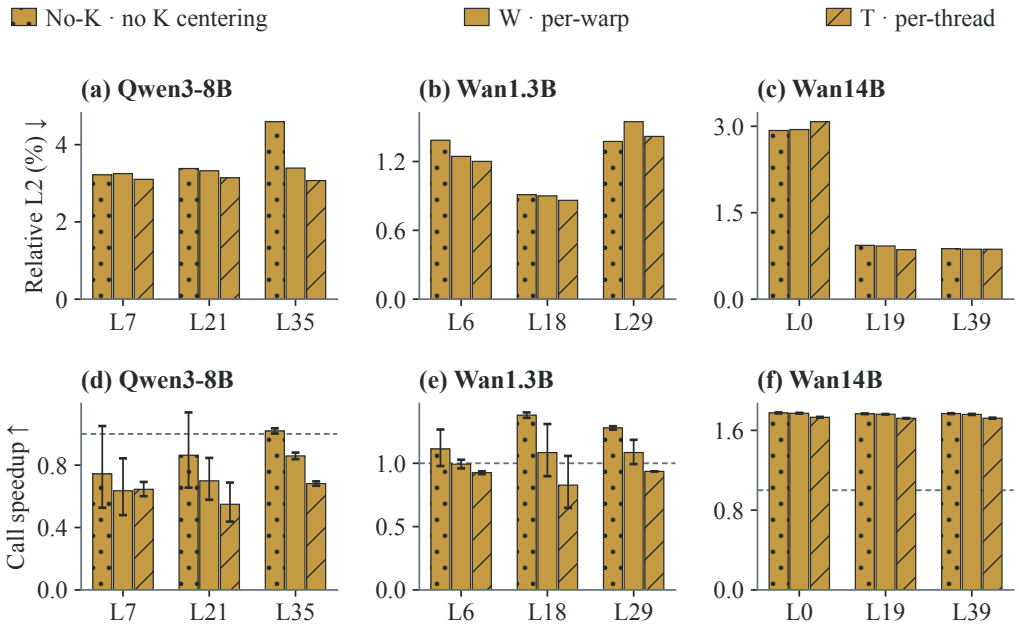


Figure 12: Sage2++ configuration controls: relative L_2 error (top) and complete-call speedup over Torch (bottom). No-K/W disable/enable K centering with per-warp quantization; T uses per-thread quantization. Whiskers span two orders.

D.4 Native Conversion and Execution Fusion

Native encoding must be isolated at fixed E3 arithmetic. Four B1/H16/D128 BHSD cases combine N2048/8192 with full/causal masks. Native and optimized software P encoding share E3 scales, native MMA, and register, stack, and shared-memory resources, with no spills. Non-P functions are unchanged. Outputs are bitwise identical; paired geometric core and complete-call time reductions are 12.0% and 10.1%.

The same four length/mask combinations extend the control to D64 and D256. Native/software outputs and native/retained-runtime outputs are bitwise identical. Core time decreases by 21.6% and 5.4%, and complete-call time by 17.4% and 4.6%, respectively. The paired implementations also preserve resources and non-P device functions. D64 has no spills; both D256 implementations retain identical compiler spill traffic, so its benefit is not attributed to spill elimination. Changed-query Graph checks, memcheck, and synccheck pass. Table 7 retains the original D128 pair and the independent D64/D256 measurements without pooling absolute times across cohorts.

Figure 5a shows four cases per head dimension and their aggregate reductions; bars transform paired geometric speedup s into time reduction $100(1 - 1/s)\%$.

Preparation and output fusion. Preparation and output fusion reduce repeated data access; their benefit depends on input size and corrections. Mean CTAs reduce disjoint sequence segments over adjacent channels; long inputs use partial and final reductions. OC combines BF16 auxiliary-buffer

writes, with independent CTAs preparing Q/K and V in the same grid on the caller's stream. FineQ omits unread causal tiles. Direct output requires complete writes and preserves Smooth-V's intermediate BF16 rounding and recovery order. Independent samples in Table 7 include twelve short-mean cells per dimension, ten FineQ cells, and 25/32/25 direct-output cells for D64/128/256. OC averages include many unresolved cells and do not imply a per-shape guarantee.

Table 7: Paired complete-call time reductions over each ablation's shape/layout/policy cells. Rows use separate unoptimized controls and are not additive. † marks reductions below 3%; a resolved cell exceeds 3% in both orders (Appendix D.4).

Change	Call-time reduction (%)			Applicability
	D64	D128	D256	
Native P encoding	17.4	10.1	4.6	Four E3 full/causal cells per dimension
Shared short means	12.5	12.4	9.1	≤ 1024 rows, ≤ 32 groups
Causal FineQ pruning	9.8	6.3	8.2	$N \geq 8192$, divisible by 1024
OC writeback/packing	6.2	2.2†	1.4†	Native NVFP4 OC
Direct BSHD output	7.2	8.2	9.7	Complete writeback and recovery

D.5 Native Correction Controls and Single-Factor Ablation

Correction controls first test consistency across captured activations, then fix execution semantics to separate grouping from schedule and denominator effects. An independent native panel covers 18 Qwen and three Wan validation states. Smooth-K extends Raw, Smooth-Q forms Basic, and other FP4 options individually extend Basic. Causal Sink-K extends corrected QK4PV8–E3; causal QK8PV4–E4 successively adds Sink-V, Smooth-K, and Smooth-V. All 423 paired numerical results repeat exactly across the two orders. Smooth-Q and both sink corrections improve every Qwen state in their respective controls; Hadamard regresses on twelve Qwen and one Wan state. Cost includes schedule, grouping, and normalization. This experiment uses a separately identified Workstation Edition GPU cohort.

Fixed execution semantics. Public options can jointly change grouping, scheduling, and denominator rules, so they are not single-factor ablations. The single-factor control changes only query-group granularity on the same layer-39 capture. Both variants use M128/N64, an unencoded-weight denominator, ordered padded-input reductions, and identical layout adaptation. Query means and compensation groups change from 128 to 16. Across 64 fixed query positions covering all heads and keys, FineQ reduces FP32-relative output RMSE by 44.5% and increases complete-call time by 12.3% (opposite execution orders, five calls per cell). Constant-V response error does not improve, so a more accurate weight row sum alone does not explain the output improvement. This control establishes the benefit of finer centering on this capture, not a video-generation benefit. Basic and FineQ have output RMSE 0.02069 and 0.01148 and complete-call time 162.88 and 182.90 ms, respectively. For $V = 6$, the post-rounding response error $\mathrm{RMS}(O/6-1)$ is 2.07% and 2.11%.

E Language-Model Accuracy, Fidelity, and Latency

E.1 Same-Configuration Quality and TTFT

The current Server experiments use the same raw all-E3 configuration as the operator comparison: libraries `86ae0ad4`/`d0a2fca2`, BF16 attention inputs, and no correction menu. Qwen3-8B/32B have 36/64 layers, Hq32/64, Hkv8, and D128. Weights, surrounding operators, KV caches, and decoding remain BF16. Counters verify every replaced prefill layer. Sage2++ fixes per-warp quantization and Smooth-K without Smooth-V; F16/F32 denote FP32+FP16/FP32+FP32 accumulation, using the official V-scale maxima 2.25/448.

Thirty-two LongBench-v2 requests are selected before inference by the retained hash/domain/length rule, with untruncated inputs spanning 11,474–31,852 tokens. Each model uses one GPU for both opposite timing orders. TTFT spans chat-template processing, tokenization, transfer, fresh-cache prefill, and the first decoded token available on CPU, after a single-token warmup. Ratios pair by

request and order. The table reports geometric-mean seconds and the maximum allocated memory at first-token delivery; model loading and later decoding are excluded.

Quality is measured separately on the identical full inputs and attention configuration: KL(BF16‖candidate) covers the final 32 teacher-forced prediction positions per request, totaling 1,024 positions per model. Suffix NLL/PPL and full-vocabulary log probabilities are retained, but these are not full-document perplexities. A separate greedy generation pass disables thinking and caps output at 128 tokens. Table 8 counts only parsed, correct, noncapped answers; capped and unparsed cases remain in the records. Qwen3-32B generation uses another Server GPU after the paired TTFT/quality phase and is not pooled into TTFT timing.

Table 8: Current all-E3 and fixed Sage2++ modes on the same 32 requests. Mean KL and TTFT use the same attention configuration and full inputs. Correct/32 excludes capped answers; GiB is peak first-token allocation.

Model	Attention	Mean KL	TTFT (s)	Correct/32	GiB
8B	Torch BF16	0.00000	1.648	14	22.84
8B	Sage2++ F16	0.00647	1.464	13	22.84
8B	Sage2++ F32	0.00738	1.448	16	22.84
8B	Free E3	0.02201	1.415	13	23.03
32B	Torch BF16	0.00000	6.161	14	74.63
32B	Sage2++ F16	0.00427	5.492	13	74.63
32B	Sage2++ F32	0.00777	5.437	14	74.63
32B	Free E3	0.01435	5.345	15	74.94

E3 has modestly shorter TTFT than either Sage2++ mode but higher mean KL on both models. Its advantage over F32 is below the 3% timing-resolution criterion in both orders. Task accuracy, distribution fidelity, and serving cost therefore remain distinct outcomes.

E.2 Standard Tasks under the Current Configuration

The independently measured 5,120-question panel fixes 1,024 questions from each of ARC-Challenge, HellaSwag, MMLU, PIQA, and WinoGrande before inference (Clark et al., 2018; Zellers et al., 2019; Hendrycks et al., 2021; Bisk et al., 2020; Sakaguchi et al., 2019). Both model sizes use the same selection, current all-E3 configuration, and Sage2++ F32 control. Inputs are at most 2K tokens. MMLU uses five-shot prompts; the other tasks use zero-shot prompts. Fixed-template continuation likelihood uses answer-character normalization for ARC/HellaSwag/PIQA, and no normalization for MMLU/WinoGrande. Each method is remeasured; no old-format task scores are substituted.

Macro accuracy weights tasks equally. Task-stratified, question-paired bootstrap uses 10,000 resamples and percentile 95% intervals. E3-minus-BF16 differences are -0.098 and -0.059 percentage points for 8B and 32B. These estimates describe the fixed balanced panel and do not establish population-wide equivalence.

E.3 Fixed Smooth-K and Matched-Format Controls

An independent current-runtime experiment applies one fixed Smooth-K transformation: FP32 mean/subtraction over the BF16 K sequence, followed by BF16 rounding. All preparation is charged to TTFT. On the same 32 Qwen3-8B requests, mean KL rises from 0.02201 to 0.02772 despite lower captured-layer relative L2 error; 18 requests have higher KL and 14 lower KL. TTFT changes from 1.397 to 1.405 s. The raw controls reproduce the original scalar KL values exactly; timing observations from the two experiments are not pooled.

Earlier matched-format E3/E4 controls retain a separate numerical configuration to isolate representation. Nine Qwen3-8B contexts of 512/2048/8192 tokens evaluate the final 32 positions. E3 improves aggregate fidelity over E4 but fails the probability gate on three contexts. These controls support the format comparison, not an equal-fidelity speedup for the current optimized policy.

Table 9: Current-configuration task accuracy (%) on 5,120 fixed questions per model, followed by paired macro-accuracy differences and 95% intervals.

Model	Attention	ARC-C	HellaSwag	MMLU	PIQA	WinoG.	Macro
8B	Torch BF16	56.93	73.54	74.61	78.22	66.89	70.04
8B	Sage2++ F32	56.93	73.54	74.80	77.93	66.02	69.84
8B	Free E3	57.13	73.05	74.51	78.61	66.41	69.94
32B	Torch BF16	61.43	82.03	82.62	82.32	72.95	76.27
32B	Sage2++ F32	61.82	82.32	82.52	82.71	72.75	76.43
32B	Free E3	61.62	82.52	82.42	82.32	72.17	76.21
E3 minus baseline: macro percentage points [paired 95% interval]							
8B	Torch BF16					-0.098 [-0.488, +0.293]	
8B	Sage2++ F32					+0.098 [-0.410, +0.586]	
32B	Torch BF16					-0.059 [-0.391, +0.273]	
32B	Sage2++ F32					-0.215 [-0.586, +0.156]	

Table 10: Independent matched-format probability controls on nine Qwen3-8B contexts. Passing requires PPL/BF16 ≤ 1.02 and mean KL ≤ 0.02; maxima are across contexts.

Attention	Max. PPL/BF16 ↓	Context-mean KL maximum ↓	Pass/9
BF16	1.0000	0.00000	Reference
Sage2++	1.0014	0.01697	9/9
QK8PV8–E4	1.0368	0.05204	2/9
QK8PV8–E3	1.0054	0.02928	6/9

E.4 GQA Adaptation and Generation Controls

The official FP4 Sage3 interface repeats Qwen's eight K/V heads to match its query heads. A BF16 replication control produces identical token IDs and zero KL at sampled positions on four fixed prompts, isolating replication from candidate arithmetic. Full prompt text, token IDs, outputs, and layer counters are retained in the supplementary materials. All responses use the same 128-token generation cap; they are qualitative examples rather than task-accuracy estimates.

F Video Fidelity and Delivery Cost

F.1 Current-Configuration Paired Videos

The current Server experiment uses the same optimized E3 libraries as the operator and language runs (`86ae0ad4/d0a2fca2`). Wan2.1-T2V-14B has 40 layers, 40 heads, D128, and 75,600 self-attention tokens. Four fixed scene/noise pairs compare Torch BF16, raw all-E3, and both fixed per-warp Sage2++ accumulation modes. Each scene runs all four methods on the same GPU with identical initial noise and the official negative prompt. The 16 videos use 720×1280 resolution, 81 frames, 16 fps, 50 UniPC steps, and guidance/shift 5. Weights, GPU-resident T5/VAE, and cross-attention remain BF16. Each native video verifies 4,000 replaced self-attention calls and 4,000 BF16 cross-attention calls.

After two-step warmup, complete delivery includes prompt encoding, denoising, VAE decoding, host RGB transfer, and H.264 output. Model loading, warmup, and evidence serialization are excluded. Each scene/method has one complete generation; scenes, not frames, are the statistical units. Delivery ratios use each scene's same-GPU Torch reference. SSIM and RGB8 MSE measure agreement with that reference, not human preference or semantic correctness.

E3 delivers faster than either Sage2++ mode on this cohort but has lower average SSIM. Its MSE exceeds Sage2++ F32 on all four scenes. The complete media, initial-noise hashes, per-scene timing, and fidelity records are retained; no confidence interval is inferred from the 81 correlated frames.

Table 11: Current Wan14B policy on four fixed scene/noise pairs: geometric-mean delivery seconds and Torch speedup, arithmetic-mean fidelity, and maximum allocated memory. F16/F32 are the fixed Sage2++ accumulation modes.

Attention	Delivery (s)	Speedup	SSIM	RGB8 MSE	GiB
Torch BF16	2232.70	1.000×	1.0000	0.00	54.22
Sage2++ F16	1638.69	1.362×	0.7882	952.36	54.22
Sage2++ F32	1615.89	1.382×	0.8096	901.10	54.22
Free E3	1557.28	1.434×	0.6922	1575.44	54.22

F.2 Independent Matched-Format Controls

The earlier E3/E4 comparison fixes quantization groups and corrections to isolate format rather than the optimized policy above. It uses two scenes with seeds 20260914/42/2026, covering BF16, Sage3, corrected FP4, E4, and E3: 30 videos, including six explicitly identified reused observations. Five timing pairs have same-GPU Torch references; an earlier train reference is used only for content. The E3/E4 pixel-MSE geometric ratio is 0.587, with two of six pairs favoring E4. These observations are not pooled with the current Server cohort.

Table 12: Independent matched-format Wan14B controls: fidelity on six scene/seed pairs (top) and delivery speed on five same-GPU Torch pairs (bottom). These are separate from the optimized-policy results in Table 11.

Scene	Seed	FP4: SSIM ↑		FP8 format
		Sage3	Free corrected	E3/E4 MSE ↓
Turntable	20260914	0.830	0.851	1.679
Turntable	42	0.769	0.856	0.199
Turntable	2026	0.840	0.850	0.358
Train	20260914	0.400	0.662	0.573
Train	42	0.474	0.577	1.104
Train	2026	0.552	0.728	0.538
Geometric E3/E4 pixel-MSE ratio (six pairs)				0.587

Attention	Delivery speedup (Torch=1) ↑	Peak allocation (GiB) ↓
BF16	1.000	54.22
Sage3	1.453	60.22
QK4PV4, corrected	1.552	54.95
QK8PV8–E4	1.257	54.22
QK8PV8–E3	1.251	54.22

F.3 Asymmetric Precision and Content Differences

Asymmetric-precision interventions separate QK and PV effects on final videos. Figure 1 comes from an independent ten-path single-input panel. E3/E4 pairs share a process and GPU; the 8/4 pair uses another GPU with remeasured BF16/Sage3 anchors. Timing differences of 0.2–0.6% do not establish an E3/E4 ordering. Frame 27 and equal-scale crops are shown without image enhancement. Full videos retain BF16 object omissions and missing cup stripes across methods; these examples are not an independent human evaluation.

Three additional fixed scene/noise pairs compare raw 4/4, 4/8, 8/4, and 8/8 E3. QK4PV8 has lower BF16-relative luminance error than QK8PV4 in every measured frequency band, but duplicates an object in one fixed frame. Pixel agreement is therefore neither semantic correctness nor evidence for a universal division of QK as structure and PV as detail.

F.4 Correction Effects on Generated Videos

Lower captured-layer error does not automatically imply closer agreement with BF16 videos. The dense video menu panel in Figure 13 contains 24 videos, pairing each candidate with its parent under fixed scene/noise and the same GPU in both execution orders. OC indices are frozen on a BF16 forward pass with another seed. Native layer calls include dtype conversion, layout, preparation, and allocation. After FineQ warmup ran out of memory, FineQ and Scale-QK use reinitialized, allocator-matched Basic controls; interrupted runs supply no paired timing. Public options include the denominator differences in Table 3.

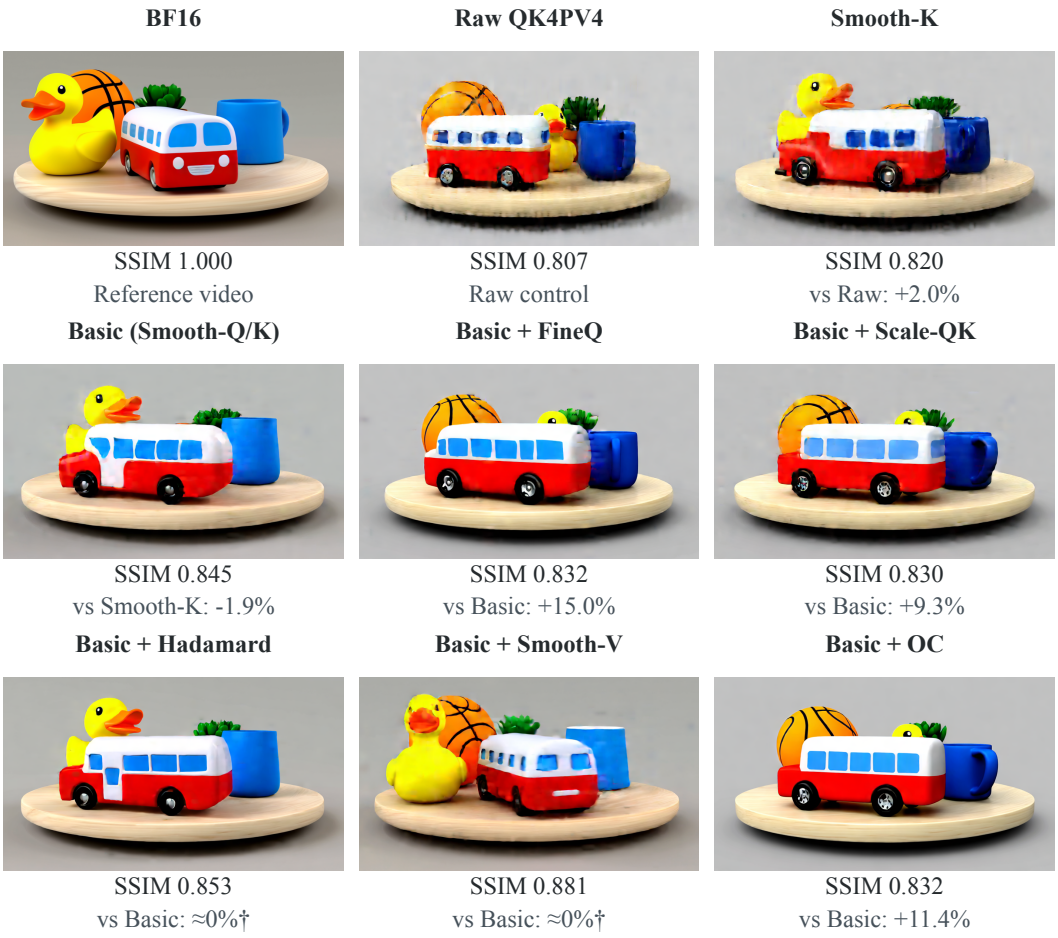


Figure 13: Wan14B FP4 correction examples, frame 27 with identical crops. SSIM covers all 81 frames in both orders. Time labels use each same-GPU Basic parent; † marks changes below 0.2% in both orders.

In the public-option comparison, FineQ's layer-39 output RMSE is $0.567\times$ Basic, but the RMS of the output error averaged over 64 fixed queries is $1.341\times$. The latter measures a shared component of output error, not Q-mean estimation error. This comparison includes denominator and schedule changes, remains separate from Appendix D.5's single-factor result, and does not establish causation during denoising. Eight further Hadamard/Smooth-V videos cover two inputs with delivery-time changes below 0.2%. Turntable fidelity changes in both directions, and one output with higher SSIM duplicates the duck.

In Figure 5b,c, points retain the three captured layers and two execution orders, respectively. Bars use geometric means of RMSE/Basic ratios and arithmetic means of paired SSIM differences. The single prompt/seed is not expanded into independent frame samples.

G INSTALLATION AND REPRODUCTION GUIDE

Code and version. `FreeAttn-repro` provides the CUDA runtime, C++/Python interfaces, instruction checks, and operator benchmarks without manuscript sources or model weights. Release `2026-09-26-family` (v0.5.0) includes the accepted E3, asymmetric-precision, and NVFP4 optimizations. `SOURCE_PROVENANCE.json` and `SOURCE_MANIFEST.json` identify its source and files. Snapshot `9f56fa3` is the earlier package; historical results retain their recorded configurations. Run commands from the package root; model and archived-result entry points are listed in `paper/REPRODUCTION.md` in the research repository.

Supported devices and requirements. FreeAttn targets NVIDIA SM120 GPUs, including GeForce RTX 5090/5080 and RTX PRO 6000 Blackwell Workstation/Server Editions. The package is validated with Linux, CUDA 13.2, and PyTorch 2.12.0+cu132, distinct from the Torch 2.14/CUDA 13.0 Server experiments. It requires Python 3.10+, CMake 3.24+, Ninja, a C++20 compiler, and NumPy; see `docs/INSTALL.md` and `baselines/README.md`.

Step 1: Install and build. In a Python environment with the required dependencies, install the interface, verify source integrity, and build the runtime:

```
export CUDA_HOME=/usr/local/cuda
python3 -m pip install -e .
python3 reproduce.py verify
python3 reproduce.py build --output results/build
```

Expected outputs are native libraries in `install/lib/` and a Tensor extension matching the active PyTorch/CUDA environment in `install/python/`. The status in `results/build/RECEIPT.json` should be `pass`.

Step 2: Validate instructions and operators. List devices with `nvidia-smi -L`, replace `GPU-YOUR-UUID` with the UUID of an idle SM120 device, and run:

```
export FA_GPU=GPU-YOUR-UUID
python3 reproduce.py instructions --gpu "$FA_GPU" \
  --memcheck --output results/instructions
python3 reproduce.py smoke --gpu "$FA_GPU" \
  --output results/smoke
python3 examples/run_demos.py --gpu "$FA_GPU" \
  --output results/demos
```

The instruction sweep should cover 2,068 matrix-instruction cases with zero memory-check errors; both run receipts should report `pass`. The six demos cover MHA/GQA, BSHD full attention, asymmetric precision, D256 tails, and CUDA Graph replay. They check immutable inputs, allocation, repeated calls, and replay against FP32 with TF32 disabled. In `results/demos/DEMOS.json`, `quality_pass` distinguishes accuracy from successful execution.

Step 3: Reproduce operator comparisons. Check the core implementation first. The `optimized` suite covers batch sizes 1 and 4 and head dimensions 64, 128, and 256. BSHD is the default; `--layout BHSD` selects the other public interface.

```
python3 reproduce.py benchmark --gpu "$FA_GPU" \
  --suite smoke --profile core --output results/core-smoke
python3 reproduce.py benchmark --gpu "$FA_GPU" \
  --suite optimized --layout BSHD --output results/core
```

Install baselines in the same Python/PyTorch/CUDA environment; mismatched native extensions are rejected.

```
python3 baselines/install_sage.py --gpu "$FA_GPU" \
  --output external/sage
python3 reproduce.py baselines --methods torch sage2 sage3
python3 reproduce.py benchmark --gpu "$FA_GPU" \
  --suite optimized --methods e3 sage2_fp32_fp32 sage3 \
  --output results/comparison
```

Use `--suite full` for the fixed 16-shape panel, and `--profile comparison` for all adapters after installing their dependencies. Each run writes `RECEIPT.json`, `benchmark/order-0.jsonl`, `benchmark/order-1.jsonl`, and `benchmark/SUMMARY.json`. Complete calls include preparation, corrections, quantization, and output allocation. Retain failures in fresh output directories; `VALIDATION.json` lists release checks.