CARE-ATTN: COMPONENT-AWARE RESIDUAL ERROR CORRECTION FOR FP4 ATTENTION QAT

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in attention quantization-aware training (QAT), notably Attn-QAT, have substantially narrowed the quality gap of FP4 attention. Nevertheless, a measurable gap to higher-precision attention remains, and end-to-end recovery alone does not reveal what QAT actually adapts to inside the attention operator. We revisit FP4 attention from a component-wise perspective and find that Attn-QAT adapts highly unevenly: it substantially reduces the error from quantized queries and keys ($\boldsymbol{QK}$), while errors from attention probabilities ($\boldsymbol{P}$) and values ($\boldsymbol{V}$) barely change and a non-negligible $\boldsymbol{QK}$ residual remains. These residuals also have distinct structures: the $\boldsymbol{QK}$ residual is sparse and heterogeneous across layers and heads, the $\boldsymbol{P}$ error is limited by the quantizer, and the $\boldsymbol{V}$ error is associated with channel-wise offsets. We therefore propose CARE-Attn (**C**omponent-**A**ware **R**esidual **E**rror Correction for Attention), which corrects each residual separately: it recomputes predicted high-error $\boldsymbol{QK}$ tiles in higher precision under adaptive budgets, decouples training and inference precision through $\boldsymbol{P}$-exact QAT, and removes channel offsets from $\boldsymbol{V}$ with exactly compensated mean-centering. On video diffusion transformers, CARE-Attn improves generation quality over Attn-QAT, produces outputs closer to those of the BF16 fine-tuned model, and reduces QAT overhead; on language models, it consistently lowers perplexity and substantially improves long-context retrieval. These results show that attention QAT alone is not enough: explicitly correcting the residuals it leaves is key to accurate FP4 attention. Code is available at `https://anonymous.4open.science/r/CARE-Attn`.

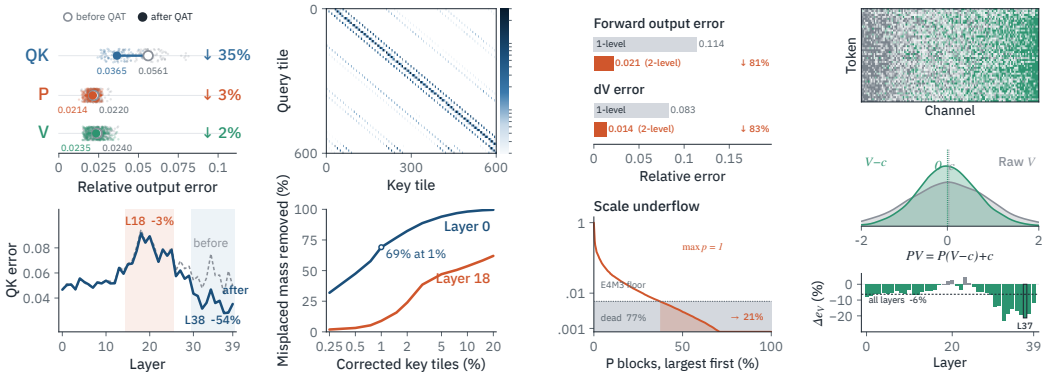
1 INTRODUCTION

Attention (Vaswani et al., 2017) is a major computational bottleneck in both diffusion models and language models, particularly at long sequence lengths. FlashAttention and its successors greatly improve attention efficiency by reorganizing computation around the GPU memory hierarchy (Dao et al., 2022; Dao, 2024; Shah et al., 2024; Zadouri et al., 2026). Meanwhile, native FP4 support on NVIDIA Blackwell GPUs provides much higher throughput than BF16 and FP8 (Zhang et al., 2025b), making FP4 attention an important step toward end-to-end low-precision execution.

Attention, however, is considerably more difficult to quantize than conventional linear layers. Its intermediate operands exhibit distinct numerical properties, and errors introduced before and after softmax propagate differently through the operator. Training-free low-precision attention therefore relies on specialized numerical treatments such as smoothing, two-level probability scaling, and fine-grained quantization (Chen et al., 2024b; Zhang et al., 2025c;a;b). Attn-QAT takes a complementary approach by training the model with FP4 attention, allowing it to adapt to quantization noise (Zhang et al., 2026b). With carefully matched forward–backward numerics, it substantially narrows the quality gap of FP4 attention. Nevertheless, a measurable gap to higher-precision attention remains in our evaluated settings, indicating that QAT does not fully eliminate the errors of FP4 computation.

This raises a more fundamental question: *what does attention QAT actually adapt to*? End-to-end recovery alone does not reveal whether QAT uniformly improves robustness to the different quantization sources inside attention, or whether some components remain numerically fragile after training. We therefore revisit FP4 attention from a component-wise perspective, separately attributing its quantization-induced output error to the $\boldsymbol{QK}$, $\boldsymbol{P}$, and $\boldsymbol{V}$ pathways. Our analysis reveals highly

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.



(a) Uneven adaptation. (b) Sparse $\boldsymbol{QK}$ residual. (c) Quantizer-limited $\boldsymbol{P}$. (d) Channel-offset $\boldsymbol{V}$.

Figure 1: **Attn-QAT leaves structured, component-specific residual errors.** Wan2.1-14B, relative attention-output error versus a high-precision reference. (a) Attn-QAT reduces $\boldsymbol{QK}$ error by 35%, while $\boldsymbol{P}$ and $\boldsymbol{V}$ change by only -3% and -2%; mid-depth layers keep most of the leftover $\boldsymbol{QK}$ residual. (b) The remaining $\boldsymbol{QK}$ error is concentrated in a few tiles, but sparsity is not uniform: a 1% high-precision budget recovers 69% of the misplaced probability mass (TV) at layer 0 versus 9% at the error-peak layer 18. (c) Two-level scaling cuts forward and $d\boldsymbol{V}$ error by about 80%, and reduces micro-blocks whose E4M3 scale underflows to zero from 77% to 21%. (d) Value error is associated with channel-wise offsets; subtracting the channel mean reduces $\boldsymbol{V}$ error by 6.3% overall.

uneven adaptation (Figure 1). On diffusion transformers, Attn-QAT substantially reduces the error induced by quantized queries and keys, while the errors from $\boldsymbol{P}$ and $\boldsymbol{V}$ change only marginally, and a non-negligible $\boldsymbol{QK}$ residual remains. More importantly, the remaining errors are not different instances of the same failure mode: the $\boldsymbol{QK}$ residual is sparse and heterogeneous across layers and heads, the $\boldsymbol{P}$ error is primarily quantizer-limited, and the $\boldsymbol{V}$ error is associated with channel-wise offsets. QAT therefore reshapes, rather than uniformly eliminates, the quantization-error landscape of FP4 attention, and these distinct residual structures call for component-aware treatments rather than a single uniform correction.

Motivated by these observations, we propose CARE-Attn (**C**omponent-**A**ware **R**esidual **E**rror Correction for Attention), which complements QAT by explicitly targeting the residual structure of each attention component. For $\boldsymbol{QK}$, we adaptively recompute a sparse subset of predicted high-error tiles in higher precision using per-(layer, head) correction budgets. For $\boldsymbol{P}$, we decouple training and inference precision: the probability operand remains unquantized during QAT, while efficient two-level FP4 quantization is retained at inference. For $\boldsymbol{V}$ with channel-wise offsets, we apply an exactly compensated channel-wise mean-centering transformation before quantization. Each correction is applied only where its corresponding residual structure is present.

Across diffusion and language models, CARE-Attn consistently improves end-to-end quality over Attn-QAT. On diffusion transformers, it additionally reduces residual FP4 attention error and QAT overhead. Together, these results show that attention QAT alone is not enough: it provides global adaptation to low-precision attention, but the structured residual errors it leaves should be corrected explicitly. Our main contributions are:

- We show that FP4 attention QAT adapts highly unevenly across attention components: on diffusion transformers, it substantially reduces $\boldsymbol{QK}$ error while leaving much of the $\boldsymbol{P}$ and $\boldsymbol{V}$ error unresolved.
- We identify distinct structures in the residual $\boldsymbol{QK}$, $\boldsymbol{P}$, and $\boldsymbol{V}$ errors and introduce CARE-Attn, combining adaptive sparse $\boldsymbol{QK}$ correction, $\boldsymbol{P}$-exact QAT, and mean-centered $\boldsymbol{V}$ quantization, each applied where its residual structure is present.
- We validate CARE-Attn on diffusion models and language models, demonstrating improved end-to-end quality over Attn-QAT together with lower residual FP4 attention error and reduced QAT overhead on diffusion transformers.

2 Background and Preliminaries

2.1 FP4 Attention

Given query, key, and value matrices $\boldsymbol{Q} \in \mathbb{R}^{N_q \times d}$ and $\boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{N_k \times d}$, scaled dot-product attention is $\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{K}^\top/\sqrt{d}$, $\boldsymbol{P} = \mathrm{softmax}(\boldsymbol{S})$, $\boldsymbol{O} = \boldsymbol{P}\boldsymbol{V}$. FlashAttention-style kernels do not materialize $\boldsymbol{P}$. Instead, for each query row i, they compute

$$m_i = \max_j S_{ij}, \qquad \widetilde{P}_{ij} = \exp(S_{ij} - m_i), \qquad \ell_i = \sum_j \widetilde{P}_{ij}, \qquad \boldsymbol{O}_{i,:} = \ell_i^{-1} \sum_j \widetilde{P}_{ij} \boldsymbol{V}_{j,:}, \quad (1)$$

so $\boldsymbol{P} = \mathrm{diag}(\boldsymbol{\ell})^{-1}\widetilde{\boldsymbol{P}}$. Low-precision fused attention therefore quantizes the numerator $\widetilde{\boldsymbol{P}}$ of the second matrix multiplication, while $\boldsymbol{\ell}$ is accumulated separately.

NVFP4 attention. NVFP4 represents values in E2M1 with hierarchical scaling: each group of 16 consecutive elements shares an E4M3 block scale, together with a higher-precision per-tensor global scale (NVIDIA, 2025; Rouhani et al., 2023; Zhang et al., 2025b; 2026b). We denote the corresponding quantize–dequantize (fake-quantization) map by $\mathcal{Q}(\cdot)$ and write $\boldsymbol{Q}^F = \mathcal{Q}(\boldsymbol{Q})$, $\boldsymbol{K}^F = \mathcal{Q}(\boldsymbol{K})$, and $\boldsymbol{V}^F = \mathcal{Q}(\boldsymbol{V})$. In the fully FP4 formulation considered in this work,

$$\widehat{\boldsymbol{S}} = \boldsymbol{Q}^F(\boldsymbol{K}^F)^\top/\sqrt{d}, \ \widehat{\widetilde{P}}_{ij} = \exp(\widehat{S}_{ij} - \widehat{m}_i), \ \widetilde{\boldsymbol{P}}^F = \mathcal{Q}_{\boldsymbol{P}}(\widehat{\widetilde{\boldsymbol{P}}}), \ \widehat{\boldsymbol{O}}_{i,:} = \widehat{\ell}_i^{-1} \sum_j \widetilde{P}^F_{ij} \boldsymbol{V}^F_{j,:}, \quad (2)$$

where $\widehat{m}_i = \max_j \widehat{S}_{ij}$, $\widehat{\ell}_i = \sum_j \widehat{\widetilde{P}}_{ij}$, and $\mathcal{Q}_{\boldsymbol{P}}$ is the quantizer applied to the probability numerator (single-level NVFP4 in Attn-QAT). Throughout the paper, $\boldsymbol{P}$ quantization means quantizing this kernel-native numerator $\widetilde{\boldsymbol{P}}$. We use *micro-block* for the 16-element NVFP4 scaling group and *tile* for a two-dimensional FlashAttention block $(\boldsymbol{Q}_I, \boldsymbol{K}_J)$ of B_q query rows and B_k key rows; the latter is the granularity of adaptive sparse $\boldsymbol{Q}\boldsymbol{K}$ correction.

2.2 Quantization-Aware Training for FP4 Attention

Quantization-aware training (QAT) simulates low-precision computation in the forward pass while retaining high-precision weights, with gradients through $\mathcal{Q}$ approximated by a straight-through estimator (Bengio et al., 2013; Jacob et al., 2018; Esser et al., 2020). In this paper, we use QAT to refer to QAT for the fused attention operator, not for linear GEMMs unless otherwise noted. Applying QAT to fused attention is non-trivial because FlashAttention recomputes softmax quantities during the backward pass.

Attn-QAT (Zhang et al., 2026b) resolves two resulting precision mismatches. First, the $\boldsymbol{P}$ tensor used for $d\boldsymbol{V}$ is recomputed and fake-quantized consistently with the forward $\boldsymbol{P}\boldsymbol{V}$ computation. Second, FlashAttention evaluates the row-wise $D_i = \langle \boldsymbol{P}_{i,:}, d\boldsymbol{P}_{i,:} \rangle = \langle d\boldsymbol{O}_{i,:}, \boldsymbol{O}_{i,:} \rangle$ without materializing the full $\boldsymbol{P}$ matrix. This identity requires the same $\boldsymbol{P}$ matrix in $\boldsymbol{O}$ and in the softmax Jacobian. Because the output $\widehat{\boldsymbol{O}}$ uses fake-quantized $\boldsymbol{P}$, substituting $\widehat{\boldsymbol{O}}$ for $\boldsymbol{O}$ breaks the equality. It therefore maintains an auxiliary output $\boldsymbol{O}'_{i,:} = \widehat{\ell}_i^{-1} \sum_j \widehat{\widetilde{P}}_{ij} \boldsymbol{V}^F_{j,:}$, used only in the backward pass, which restores $D_i = \langle d\boldsymbol{O}_{i,:}, \boldsymbol{O}'_{i,:} \rangle$. These modifications enable stable FP4 attention QAT.

3 Uneven Error Adaptation in FP4 Attention QAT

Although Attn-QAT narrows the quality gap of FP4 attention, a measurable gap remains, and end-to-end quality does not reveal how QAT adapts to different quantization sources within attention. We therefore analyze FP4 attention directly and ask two questions: which component-wise errors are reduced by QAT, and what residual quantization errors remain after training?

We present the analysis on Wan2.1-14B; the corresponding analysis for language models is in Appendix A.6. Following Section 2.1, we isolate the $\boldsymbol{Q}\boldsymbol{K}$, $\boldsymbol{P}$, and $\boldsymbol{V}$ pathways by enabling only one quantization source at a time while keeping the remaining computation in high precision. For $X \in \{\boldsymbol{Q}\boldsymbol{K}, \boldsymbol{P}, \boldsymbol{V}\}$, we define the component-wise relative output error as

$$e_X = \mathbb{E}_i \left[\frac{\left\| \boldsymbol{O}^X_{i,:} - \boldsymbol{O}_{i,:} \right\|_2}{\left\| \boldsymbol{O}_{i,:} \right\|_2} \right], \quad (3)$$

where $\boldsymbol{O}$ is the high-precision attention output and $\boldsymbol{O}^X$ is the output with only component X quantized. We additionally report e_{all} with all three sources enabled simultaneously. Because these perturbations interact through softmax and value aggregation, $e_{\boldsymbol{QK}}$, $e_{\boldsymbol{P}}$, and $e_{\boldsymbol{V}}$ are used for error attribution rather than as an additive decomposition of e_{all}.

Attention QAT exhibits highly uneven component-wise adaptation. Figure 1a shows that Attn-QAT substantially reduces $e_{\boldsymbol{QK}}$, whereas $e_{\boldsymbol{P}}$ and $e_{\boldsymbol{V}}$ change only marginally. This indicates that the reduction in overall FP4 attention error comes mainly from adaptation of the $\boldsymbol{QK}$ pathway rather than from a uniform increase in robustness across all quantized components. Importantly, a non-negligible $\boldsymbol{QK}$ residual remains afterward. Thus, QAT reshapes the component-wise error distribution but does not uniformly eliminate residual quantization error.

Residual $\boldsymbol{QK}$ error is sparse and heterogeneous. We next examine how the remaining $\boldsymbol{QK}$ error is distributed across the attention computation. For each query row, we measure the total variation (TV) between the high-precision and $\boldsymbol{QK}$-quantized attention distributions. Because TV decomposes additively over disjoint key tiles (Appendix A.1), it allows us to attribute the residual probability error to individual FlashAttention tiles.

We find that the residual is strongly concentrated: a small fraction of attention tiles accounts for most of the $\boldsymbol{QK}$-induced probability error, and recomputing these tiles in high precision substantially reduces the residual (Figure 1b). The concentration, however, is highly heterogeneous across (layer, head) pairs. Heads within the same layer can range from nearly uniform residual error to extremely sparse concentration (Figure 3a). Moreover, this (layer, head)-specific pattern is preserved across denoising steps (Appendix A.1). We evaluate how 1% sparse recovery (maintaining 1% of key tiles per query tile in high precision) and QAT act on different layers (Figure 6). QAT reduces $e_{\boldsymbol{QK}}$ mainly after layer 26 (layer 38: -54%), while 1% sparse recovery is spread across depth, including the mid-depth error peak that QAT barely reduces (layer 18: -3%). These observations indicate that the natural unit for correction is the (layer, head) pair. This motivates *adaptive sparse* $\boldsymbol{QK}$ *correction* with a per-(layer, head) correction budget.

Residual $\boldsymbol{P}$ error is quantizer-limited. The error induced by quantizing $\widetilde{\boldsymbol{P}}$ changes marginally after QAT. The residual error primarily comes from the probability quantizer. Because the softmax numerator is highly concentrated near zero, single-level FP4 quantization suffers from poor scale utilization and underflow of small probability values (Figure 1c). The same probability quantization error also propagates to the backward pass because the fake-quantized probabilities are reused to compute $d\boldsymbol{V}$. Switching to two-level scaling (Section 4.2) cuts both forward and backward error by about 80% (Figure 1c), indicating that the residual $\boldsymbol{P}$ error is primarily *quantizer-limited*.

Residual $\boldsymbol{V}$ error carries channel-wise offsets. The error induced by quantizing $\boldsymbol{V}$ likewise changes only marginally after QAT. The value activations exhibit channel-wise offsets. These offsets consume a substantial fraction of the FP4 dynamic range and reduce the resolution available for token-dependent variation. Removing the channel-wise mean before quantization reduces the residual output error, indicating that part of the residual $\boldsymbol{V}$ error is *offset-induced* (Figure 1d).

4 CARE-ATTN

CARE-Attn applies a different treatment to each residual source identified in Section 3: adaptive sparse $\boldsymbol{QK}$ correction, $\boldsymbol{P}$-exact QAT, and mean-centered $\boldsymbol{V}$ quantization (Figure 2).

4.1 ADAPTIVE SPARSE $\boldsymbol{QK}$ CORRECTION

Section 3 shows that the residual $\boldsymbol{QK}$ error after QAT is sparse and highly heterogeneous across layers and heads (Figure 3a). We therefore propose *adaptive sparse* $\boldsymbol{QK}$ *correction*, which separates correction into two stages. An offline stage assigns each (layer, head) a correction budget according to its residual-error profile, while an online stage selects the specific tiles to be recomputed in high precision for the current input. Both stages target the recovery of total-variation (TV) error induced by $\boldsymbol{QK}$ quantization.

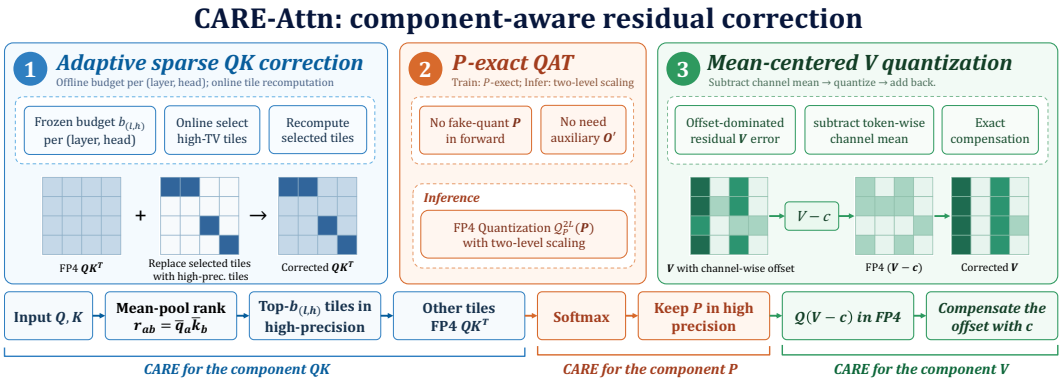


Figure 2: **Overview of CARE-Attn.** Each residual error from Figure 1 is corrected separately, while uncorrected computation stays on the FP4 path.

TV-guided correction objective. Let $\boldsymbol{P}$ and $\widehat{\boldsymbol{P}}$ denote the attention probabilities computed from high-precision and quantized $\boldsymbol{Q}, \boldsymbol{K}$, respectively, with all other operands kept in high precision. For query row i, we define

$$\mathrm{TV}_i = \frac{1}{2} \sum_j \left| P_{ij} - \widehat{P}_{ij} \right|. \tag{4}$$

TV isolates the $\boldsymbol{QK}$ pathway from errors introduced by quantized $\boldsymbol{V}$ and decomposes additively over disjoint key tiles. We therefore use recovered TV as the common objective for both budget allocation and online tile selection.

Offline budget allocation. Let $\mathcal{U}$ be the set of layer–head pairs $u = (\ell, h)$, and let the correction budget $b \in [0, 1]$ be the fraction of key tiles recomputed in high precision. Write tv_u^0 for its residual TV without correction and $\mathrm{tv}_u(b)$ for the TV under budget b. Given an average budget B, we solve

$$\max_{\{b_u\}} \sum_{u \in \mathcal{U}} \left[\mathrm{tv}_u^0 - \mathrm{tv}_u(b_u) \right] \qquad \text{s.t.} \qquad \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} b_u \le B. \tag{5}$$

This objective favors units that both incur substantial residual error and can recover that error efficiently with additional high-precision tiles. We estimate discrete recovery curves on calibration data and optimize their upper concave envelopes under the global budget constraint. The resulting per-(layer, head) budget table is calibrated once and then frozen. At a 1% average budget, this allocation raises TV recovery from 17.0% under a uniform budget to 26.7% (Figure 3c). Details of the discrete budget levels and allocation algorithm are provided in Appendix A.2.

Online tile selection. The offline budget specifies *how many* tiles to correct, while the tiles themselves are selected from the current input. For query tile I and key tile J, we mean-pool the corresponding representations,

$$\bar{\boldsymbol{q}}_I = \frac{1}{B_q} \sum_{i \in I} \boldsymbol{Q}_{i,:}, \qquad \bar{\boldsymbol{k}}_J = \frac{1}{B_k} \sum_{j \in J} \boldsymbol{K}_{j,:}, \tag{6}$$

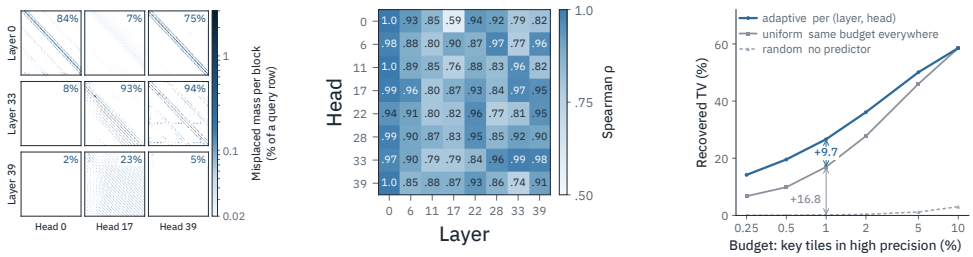
and compute the compressed score

$$r_{IJ} = \bar{\boldsymbol{q}}_I \cdot \bar{\boldsymbol{k}}_J. \tag{7}$$

For each query tile, we independently select the top-ranked key tiles according to the calibrated budget b_u. The mean-pooled score is strongly rank-correlated with tile-wise TV contribution (mean Spearman $\rho = 0.886$; Figure 3b), so the inexpensive ranking recovers the tiles that actually carry the residual. The predictor operates on one representative vector per tile and is therefore negligible compared with the full GEMM ($1/(B_q B_k)$ of its arithmetic, with $B_q = B_k = 128$ in our kernels).

Sparse high-precision correction. Let $M_{IJ} \in \{0, 1\}$ denote the resulting correction mask. The default FP4 score tile is

$$\widehat{\boldsymbol{S}}_{IJ}^{\mathrm{FP4}} = \frac{\mathcal{Q}(\boldsymbol{Q}_I)\mathcal{Q}(\boldsymbol{K}_J)^\top}{\sqrt{d}}, \tag{8}$$



(a) $\boldsymbol{QK}$ sparsity. (b) Tile selector. (c) Budget recovery.

Figure 3: **Adaptive sparse $\boldsymbol{QK}$ correction.** (a) The number in each panel denotes the fraction of TV concentrated in the top 1% of key tiles. (b) The mean-pooled tile score is strongly rank-correlated with tile-wise TV contribution (mean Spearman $\rho = 0.886$). (c) At a 1% average budget, per-(layer, head) allocation improves TV recovery from 17.0% under uniform allocation to 26.7%.

and selected tiles are recomputed using high-precision operands:

$$\widehat{\boldsymbol{S}}_{IJ} = (1 - M_{IJ})\widehat{\boldsymbol{S}}^{\mathrm{FP4}}_{IJ} + M_{IJ}\frac{\boldsymbol{Q}_I\boldsymbol{K}_J^\top}{\sqrt{d}}. \tag{9}$$

The corrected logits then enter the original online-softmax pipeline. The same correction mask and mixed-precision computation are used in the forward pass and all backward logit recomputations, preserving forward–backward consistency during QAT.

4.2 $\boldsymbol{P}$-Exact QAT

Section 3 shows that the residual $\boldsymbol{P}$ error is primarily quantizer-limited. Similar to SageAttention3 (Zhang et al., 2025b), we therefore use two-level scaling for deployment. It first applies a row-wise high-precision rescaling to the highly skewed softmax numerator before native NVFP4 microscaling, improving scale utilization and reducing the underflow of small probability values. We denote this deployment quantizer by $\mathcal{Q}^{\mathrm{2L}}_{\boldsymbol{P}}$; its detailed formulation is given in Appendix A.4.

Decoupling training and inference precision. Since QAT exhibits little adaptation to the $\boldsymbol{P}$ pathway while two-level scaling already substantially reduces its deployment error, we do not fake-quantize the softmax numerator during QAT. Instead, we compute

$$\boldsymbol{O}^{\mathrm{train}}_{i,:} = \frac{1}{\widehat{\widehat{\ell}}_i}\sum_j \widehat{\widehat{P}}_{ij}\boldsymbol{V}^F_{j,:}, \tag{10}$$

while retaining two-level FP4 probability quantization at inference:

$$\text{Train: } \widehat{\widehat{\boldsymbol{P}}} \text{ (unquantized)}, \quad \text{Inference: } \mathcal{Q}^{\mathrm{2L}}_{\boldsymbol{P}}(\widehat{\widehat{\boldsymbol{P}}}) \text{ (two-level NVFP4)}. \tag{11}$$

Here, $\boldsymbol{P}$-exact refers only to removing probability fake quantization during QAT. The logits can still be produced by low-precision $\boldsymbol{QK}$ computation. The same unquantized probability weights are used to compute $d\boldsymbol{V}$, avoiding the gradient distortion introduced by quantized probabilities.

Eliminating the auxiliary output. $\boldsymbol{P}$-exact QAT also simplifies the Attn-QAT backward path. When the forward pass uses quantized probabilities, Attn-QAT must maintain an auxiliary output $\boldsymbol{O}'$ computed with unquantized probabilities. Under equation 10, however, $\boldsymbol{O}^{\mathrm{train}} \equiv \boldsymbol{O}'$, so the additional $\widetilde{\boldsymbol{P}}\boldsymbol{V}$ accumulation is no longer required, which reduces the cost of each training step.

4.3 Mean-Centered $\boldsymbol{V}$ Quantization

Section 3 shows that part of the residual $\boldsymbol{V}$ error comes from channel-wise offsets. We therefore subtract the per-channel mean over tokens before quantization,

$$\boldsymbol{c} = \frac{1}{N_k}\sum_{j=1}^{N_k}\boldsymbol{V}_{j,:}, \qquad \boldsymbol{V}^F_{j,:} = \mathcal{Q}(\boldsymbol{V}_{j,:} - \boldsymbol{c}). \tag{12}$$

This allows the FP4 dynamic range to better represent token-dependent variation.

Under $\boldsymbol{P}$-exact QAT, the normalized attention weights $A_{ij} = \widehat{\widetilde{P}}_{ij}/\widehat{\ell}_i$ satisfy $\sum_j A_{ij} = 1$, so the removed mean can be restored exactly:

$$\widehat{\boldsymbol{O}}^{\text{train}}_{i,:} = \sum_j A_{ij}\,\mathcal{Q}(\boldsymbol{V}_{j,:} - \boldsymbol{c}) + \boldsymbol{c}. \tag{13}$$

Thus, mean-centering itself leaves the underlying attention operator unchanged; only the centered values are quantized. At inference, quantized probabilities need not sum exactly to one. Let $\alpha_i = \sum_j \widetilde{P}^F_{ij}/\widehat{\ell}_i$. We then compute

$$\widehat{\boldsymbol{O}}^{\text{infer}}_{i,:} = \frac{1}{\widehat{\ell}_i}\sum_j \widetilde{P}^F_{ij}\,\mathcal{Q}(\boldsymbol{V}_{j,:} - \boldsymbol{c}) + \alpha_i \boldsymbol{c}, \tag{14}$$

which exactly compensates for the removed offset. The gradient of mean-centered $\boldsymbol{V}$ is derived in Appendix A.5.

5 Experiments

We evaluate CARE-Attn mainly on diffusion models and verify its generality on language models, focusing on three questions: end-to-end quality over Attn-QAT, the contribution of each correction, and the training and inference cost. Full configurations and additional results are in Appendix B.

5.1 Experimental Setup

Models. For diffusion models, we train Wan2.1-14B (Wan et al., 2025) for 400 steps with the same recipe as Attn-QAT and evaluate VBench (Huang et al., 2024) and fidelity to the BF16 fine-tune (PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018) on videos from the same prompt and seed); Wan2.1-1.3B results are in Appendix B.3. For language models, we additionally train Qwen3-8B, Qwen3-14B (Yang et al., 2025), Llama-3.1-8B (Grattafiori et al., 2024), and Mistral-7B-v0.3 (Jiang et al., 2023) on C4 (Raffel et al., 2020) at sequence length 8192 following the Attn-QAT setting, and report perplexity on held-out C4 and WikiText-2 (Merity et al., 2017).

Baselines and configurations. The primary baseline is Attn-QAT trained with the same recipe (for Wan2.1-14B, we compare to its released checkpoint; a reproduced run is compared in Appendix B.3). We also report BF16 models, the training-free SageAttention3 (Zhang et al., 2025b) for video, and Attn-QAT + corr., which applies our inference-side corrections to Attn-QAT weights without retraining.

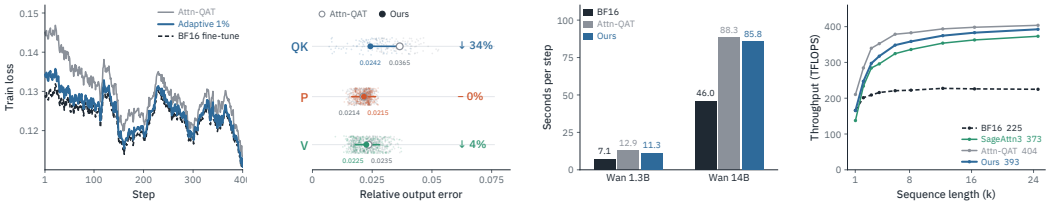
Instantiation. Each correction is used where its residual structure is present. Both model families use $\boldsymbol{P}$-exact QAT with two-level $\boldsymbol{P}$ and sparse $\boldsymbol{QK}$ correction, since $\boldsymbol{QK}$ remains the largest error after QAT and $\boldsymbol{P}$ is quantizer-limited in language models too (Appendix A.6). Video adds mean-centered $\boldsymbol{V}$ and an adaptive $\boldsymbol{QK}$ budget of 1% of key tiles on average. Language models keep $\boldsymbol{V}$ raw, as their $\boldsymbol{V}$ error comes from attention-sink outliers (Xiao et al., 2024) and a mean over all value tokens would depend on future tokens and change as the KV cache grows. They give every head three key tiles per query tile, as the first tile already recovers 60%–71% of the perplexity gain of computing all $\boldsymbol{QK}$ tiles in high precision.

5.2 End-to-End Results

Diffusion transformers. Table 1 reports VBench results on Wan2.1-14B. Attn-QAT lowers the weighted score of the original BF16 model from 0.8442 to 0.8390, mostly through dynamic degree ($0.4908 \rightarrow 0.3926$). CARE-Attn raises the weighted score to 0.8472, with gains in imaging quality ($+0.020$), aesthetic quality ($+0.008$), and dynamic degree ($+0.068$). It is the best FP4 method in the weighted score and comes within 0.0014 of the BF16 fine-tune, while exceeding it in imaging quality. Its videos are also much closer to those of the BF16 fine-tune: PSNR rises from 11.34 to 15.76 dB and LPIPS drops on 99% of prompts (see details in Table 3 and Appendix B.4), and these gains hold against a reproduced Attn-QAT run (Appendix B.3). CARE-Attn trains stably (Figure 4a) and, under the component-wise measurement of Figure 1, lowers $e_{\boldsymbol{QK}}$ by 34% and $e_{\boldsymbol{V}}$ by 4% from Attn-QAT (Figure 4b).

Table 1: **VBench results on Wan2.1-14B** (326 prompts; higher is better). Weighted = (six dimensions + 0.5 × dynamic degree)/6.5. Attn-QAT uses the official checkpoint of Zhang et al. (2026b). Bold marks the best FP4 method in each column.

Method	Imaging	Aesthetic	Subject	Background	Flicker	Smoothness	Dynamic	Weighted
BF16 (original)	0.6931	0.6670	0.9569	0.9624	0.9754	0.9871	0.4908	0.8442
BF16 fine-tune	0.7001	0.6780	0.9668	0.9708	0.9783	0.9887	0.4663	0.8486
SageAttention3	0.6723	0.6517	0.9528	0.9584	0.9756	0.9859	0.4509	0.8342
Attn-QAT	0.6823	0.6680	**0.9671**	**0.9700**	**0.9809**	**0.9888**	0.3926	0.8390
CARE-Attn	**0.7021**	**0.6760**	0.9654	0.9687	0.9768	0.9879	**0.4601**	**0.8472**



(a) Training loss. (b) Component-wise error. (c) Training speed. (d) Inference throughput.

Figure 4: **Training dynamics, residual error, and cost on Wan2.1.** (a) Training loss on Wan2.1-14B (31-step moving average). (b) Relative attention-output error, measured as in Figure 1. (c) Seconds per training step; the 1.3B CARE-Attn point uses a uniform 1% $\boldsymbol{QK}$ budget. (d) Attention kernel throughput on one RTX PRO 5000.

Large language models. Table 2 summarizes the four language models. CARE-Attn is the best FP4 method on every model and on both C4 and WikiText-2 (Appendix B.2), and it closes 71%–87% of the C4 gap between Attn-QAT and BF16. On long-context retrieval with three corrected $\boldsymbol{QK}$ tiles, it beats Attn-QAT in all 10 settings from 16k to 64k tokens, by up to 46.0 exact-match points (Appendix B.2).

5.3 Ablation Studies

Contribution of each correction. Table 3 adds the corrections one at a time on Wan2.1-14B, and correcting the $\boldsymbol{P}$ pathway gives the largest step (e_{all} 0.1132 → 0.0418, weighted VBench 0.8390 → 0.8464, PSNR 11.34 → 14.53 dB). Mean-centered $\boldsymbol{V}$ and adaptive $\boldsymbol{QK}$ correction then reduce e_{all} to 0.0318 at a 1% budget (0.0300 at 2%), 72% below Attn-QAT, and add another 1.2 dB, with the full method reaching 0.8472 on VBench. The $\boldsymbol{P}$ error is set mainly by the inference quantizer: on the same weights, two-level scaling gives $e_{\boldsymbol{P}} = 0.021$ versus 0.114 with single-level scaling, and training with two-level fake-quantized $\boldsymbol{P}$ matches $\boldsymbol{P}$-exact QAT (0.8460 vs. 0.8464) at 28% higher step time. On Qwen3-8B, two-level $\boldsymbol{P}$ alone raises the C4 perplexity of Attn-QAT by 0.029, whereas together with $\boldsymbol{P}$-exact QAT it lowers it by 0.053, and by 0.075 with training-time $\boldsymbol{QK}$ correction added (Appendix B.2).

Training with the corrections versus adding them afterward. The inference-side corrections could instead be added to Attn-QAT weights after training, which is the Attn-QAT + corr. baseline. CARE-Attn outperforms it on every language model, and on Qwen3-8B, Attn-QAT weights with eight corrected tiles per query tile still trail CARE-Attn weights with one tile (C4 11.372 vs. 11.354). On Wan2.1-14B, adding the corrections to reproduced Attn-QAT weights lowers PSNR by 0.83 dB, whereas training with them under the same inference setting raises it by 3.69 dB (Appendix B.3). Without any training, the corrections match Attn-QAT only with a 10% $\boldsymbol{QK}$ budget (0.8392 vs. 0.8390), whereas CARE-Attn reaches 0.8472 with 1%.

5.4 Efficiency Analysis

We measure cost on Wan2.1. By removing the auxiliary $\widetilde{\boldsymbol{PV}}$ accumulation of Attn-QAT, $\boldsymbol{P}$-exact QAT alone cuts step time by 19.8% on Wan2.1-1.3B and 17.6% on Wan2.1-14B (Appendix B.1).

Table 2: Average perplexity on C4 and WikiText-2 at length 8192 (↓). +corr.: Attn-QAT + $\boldsymbol{QK}$ corr.; BF16: BF16 fine-tune.

Model	Attn-QAT	+corr.	CARE-Attn	BF16
Qwen3-8B	9.05	8.93	**8.87**	8.79
Llama-3.1-8B	7.77	7.62	**7.56**	7.50
Mistral-7B	6.58	6.53	**6.52**	6.51
Qwen3-14B	8.25	8.17	**8.13**	8.08

Table 3: **Ablation on Wan2.1-14B.** e_{all}: equation 3 over all heads of layer 37. PSNR/LPIPS: vs. BF16 fine-tune.

Method	e_{all}↓	PSNR↑	LPIPS↓
Attn-QAT	0.1132	11.34	0.543
$\boldsymbol{P}$-exact QAT	0.0418	14.53	0.369
+ $\boldsymbol{V}$ centering	0.0375	15.10	0.348
+ $\boldsymbol{QK}$ corr. (CARE-Attn)	**0.0318**	**15.76**	**0.315**

With all three corrections, CARE-Attn remains faster than Attn-QAT: 11.3 versus 12.9 s per step on Wan2.1-1.3B and 85.8 versus 88.3 s on Wan2.1-14B (Figure 4c), with nearly all of the added cost from $\boldsymbol{QK}$ correction. At inference, it keeps 97% of the Attn-QAT kernel throughput at 24k tokens (393 vs. 404 TFLOPS) and exceeds SageAttention3 (373) and BF16 (225) (Figure 4d).

6 Related Work

Low-precision attention. FlashAttention-3 explores FP8 attention (Shah et al., 2024), while the SageAttention series progressively reduces attention precision to FP4 (Zhang et al., 2025c;a;b). INT-FlashAttention similarly studies integer quantization within fused attention kernels (Chen et al., 2024b). SageAttention3 combines $\boldsymbol{K}$ and $\boldsymbol{Q}$ smoothing with two-level scaling of the softmax numerator.

Quantization-aware training for attention. QAT simulates quantization during training and typically uses straight-through estimators for backpropagation (Bengio et al., 2013; Jacob et al., 2018; Esser et al., 2020). Most post-training quantization and QAT methods for LLMs and diffusion transformers target linear layers (Frantar et al., 2023; Lin et al., 2024; Liu et al., 2023; Chen et al., 2024a). Most closely related to our work, Attn-QAT enables FP4 attention QAT by maintaining forward–backward numerical consistency and the auxiliary high-precision output required by FlashAttention backward (Zhang et al., 2026b). SageBwd further shows that low-precision sensitivity can differ across attention backward pathways (Zhang et al., 2026a). While Attn-QAT narrows the quality gap, we ask which errors it adapts to and which remain, and find highly uneven adaptation across $\boldsymbol{QK}$, $\boldsymbol{P}$, and $\boldsymbol{V}$ on diffusion transformers.

Numerical transformations and sparse correction. Low-precision methods often improve quantization robustness through numerical transformations, including scaling in SmoothQuant (Xiao et al., 2023) and rotations in QuaRot (Ashkboos et al., 2024) and SpinQuant (Liu et al., 2025). SageAttention2 optionally subtracts the token mean of $\boldsymbol{V}$ to protect its FP22 accumulator (Zhang et al., 2025a); we apply this translation to FP4 quantization of $\boldsymbol{V}$ inside QAT, with exact compensation for quantized probabilities. Our adaptive sparse $\boldsymbol{QK}$ correction is most closely related to SpargeAttention (Zhang et al., 2025d), which predicts attention blocks from compressed queries and keys. Sparse attention exploits sparsity in the attention map and skips low-importance blocks to reduce computation (Jiang et al., 2024; Gao et al., 2024; Yuan et al., 2025). In contrast, we exploit sparsity in the *quantization error* and recompute a subset of high-error tiles in high precision.

7 Conclusion

In this work, we revisit FP4 attention QAT from a component-wise perspective and show that it leaves structured residual errors in the $\boldsymbol{QK}$, $\boldsymbol{P}$, and $\boldsymbol{V}$ pathways, with highly uneven adaptation. Targeting these structures, we introduce CARE-Attn, which combines adaptive sparse $\boldsymbol{QK}$ correction, $\boldsymbol{P}$-exact QAT, and mean-centered $\boldsymbol{V}$ quantization. Together, these mechanisms complement attention QAT, reducing the residual numerical error while improving the efficiency and accuracy of FP4 attention on video diffusion transformers, with consistent quality gains on language models. Our results suggest that effective low-precision attention requires not only end-to-end adaptation, but also explicit treatment of the component-specific errors that attention QAT leaves unresolved.

AI USE STATEMENT

In this work, we used generative AI tools for assistance with refining hypotheses and parts of the conceptual framework, feedback on the experimental design, helping implement parts of the methods and evaluation code, and cross-checking the interpretation of results against the raw experiment logs. We have not used generative AI tools for formulating or proving mathematical claims. We have not used generative AI tools for synthetic-data generation, translation, dataset cleaning, or qualitative evaluation. Additionally, we used generative AI tools for editing and drafting parts of the paper, formatting tables and figures, writing plotting and analysis scripts, and searching the literature. We have reviewed all AI-assisted work. All AI suggestions were discussed and validated by the authors, who made the final decisions on the method, the experiments, and the reported results; AI-assisted code was reviewed and tested, and all reported numbers were recomputed from the raw experiment logs. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the accuracy and efficiency of low-precision attention and does not involve human subjects, personal data, or the collection of new datasets. The models and datasets used in our experiments are publicly available and are used in accordance with their respective licenses and usage terms. We do not introduce new task capabilities beyond those of the underlying models; rather, our method targets the numerical accuracy and efficiency of their attention computation. As with the original models, downstream use remains subject to their existing licenses, safeguards, and usage policies.

REPRODUCIBILITY STATEMENT

Section 4 and Appendix A describe every component of CARE-Attn, including the budget allocation algorithm, the tile selector, the two-level $\boldsymbol{P}$ quantizer, and the backward pass under $\boldsymbol{P}$-exact QAT. Appendix B.1 lists the training and sampling settings for all models (Table 5), including data, optimizer, batch size, number of steps, seeds, and hardware, together with the fidelity and perplexity protocols; the long-context retrieval protocol is given in Appendix B.2. The Attn-QAT baseline for Wan2.1-14B is the publicly released checkpoint. Our code is available at `https://anonymous.4open.science/r/CARE-Attn`.

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*, 2024.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Lei Chen, Yuan Meng, Chen Tang, Xinzhu Ma, Jingyan Jiang, Xin Wang, Zhi Wang, and Wenwu Zhu. Q-DiT: Accurate post-training quantization for diffusion transformers. *arXiv preprint arXiv:2406.17343*, 2024a.

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*, 2024b.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, 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*, 2022.

Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. In *International Conference on Learning Representations*, 2020.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations*, 2023.

Yizhao Gao, Zhichen Zeng, Dayou Du, Shijie Cao, Peiyuan Zhou, Jiaxing Qi, Junjie Lai, Hayden Kwok-Hay So, Ting Cao, Fan Yang, and Mao Yang. SeerAttention: Learning intrinsic sparse attention in your LLMs. *arXiv preprint arXiv:2410.13276*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *Conference on Language Modeling*, 2024.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In *Advances in Neural Information Processing Systems*, 2024.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, 2024.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. LLM-QAT: Data-free quantization aware training for large language models. *arXiv preprint arXiv:2305.17888*, 2023.

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*, 2025.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

NVIDIA. Pretraining large language models with NVFP4. *arXiv preprint arXiv:2509.25149*, 2025.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

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*, 2023.

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*, 2024.

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.

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

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

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 *Proceedings of the 40th International Conference on Machine Learning*, pp. 38087–38099, 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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Y. X. Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. Native sparse attention: Hardware-aligned and natively trainable sparse attention. *arXiv preprint arXiv:2502.11089*, 2025.

Ted Zadouri, Markus Hoehnerbach, Jay Shah, Vijay Thakkar, and Tri Dao. FlashAttention-4: Algorithm and kernel pipelining co-design for asymmetric hardware scaling. In *Proceedings of Machine Learning and Systems*, 2026.

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

Jintao Zhang, Jia Wei, Haoxu Wang, Pengle Zhang, Xiaoming Xu, Haofeng Huang, Kai Jiang, Jianfei Chen, and Jun Zhu. SageAttention3: Microscaling FP4 attention for inference and an exploration of 8-bit training. In *Advances in Neural Information Processing Systems*, 2025b.

Jintao Zhang, Jia Wei, Pengle Zhang, Jun Zhu, and Jianfei Chen. SageAttention: Accurate 8-bit attention for plug-and-play inference acceleration. In *International Conference on Learning Representations*, 2025c.

Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. SpargeAttention: Accurate and training-free sparse attention accelerating any model inference. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 76397–76413, 2025d.

Jintao Zhang, Marco Chen, Haoxu Wang, Kai Jiang, Ion Stoica, Joseph E. Gonzalez, Jianfei Chen, and Jun Zhu. SageBwd: A trainable low-bit attention. *arXiv preprint arXiv:2603.02170*, 2026a.

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*, 2026b.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Overview of the Appendix

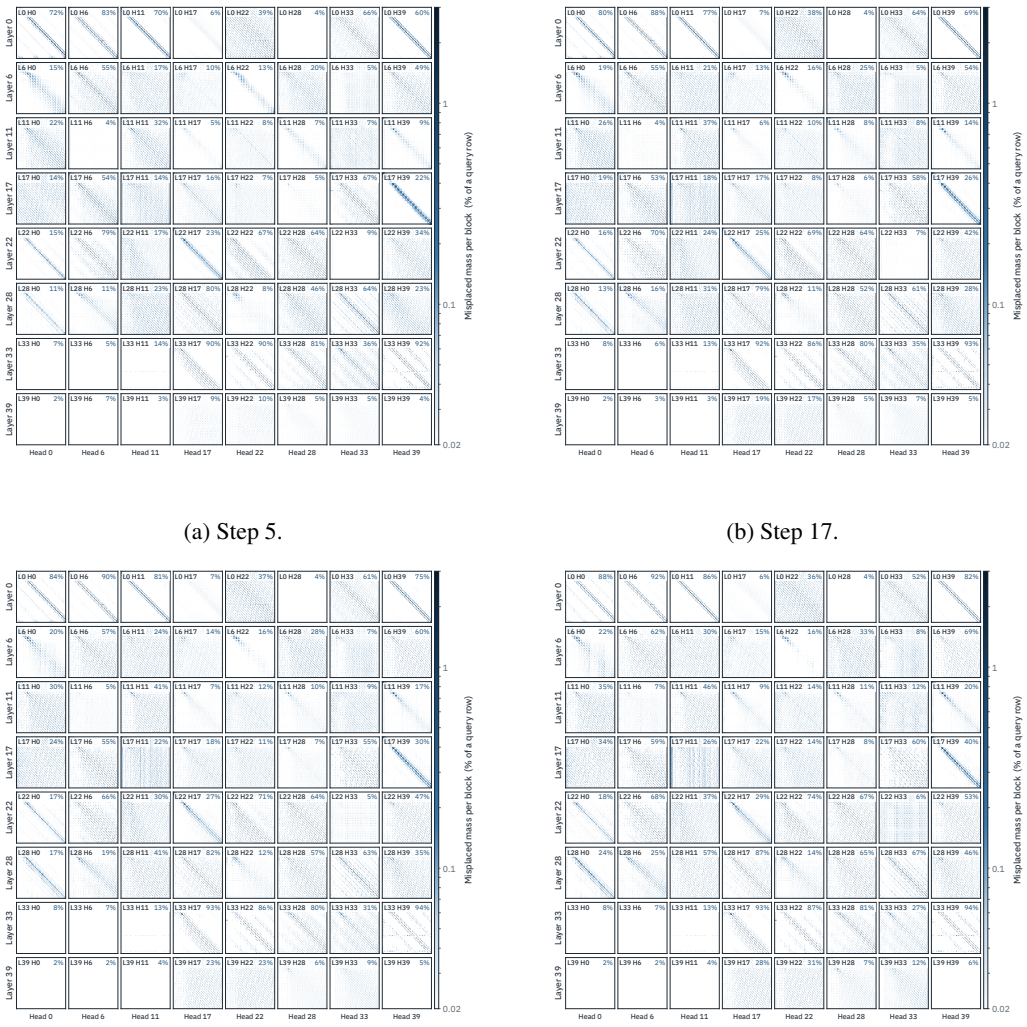
Appendix A details the analysis and components of CARE-Attn: the tile-wise $\boldsymbol{QK}$ residual measurement behind Section 3 (Appendix A.1), budget allocation and tile selection for $\boldsymbol{QK}$ correction (Appendices A.2–A.3), the two-level $\boldsymbol{P}$ quantizer and the backward pass of $\boldsymbol{P}$-exact QAT (Appendix A.4), the gradient of mean-centered $\boldsymbol{V}$ (Appendix A.5), and the residual structure of language models, which determines the corrections they use (Appendix A.6). Appendix B gives the training and evaluation settings (Appendix B.1), additional language-model results (Appendix B.2), additional diffusion results, including fidelity and Wan2.1-1.3B (Appendix B.3), and visual comparisons (Appendix B.4).

A Additional Details and Analysis of CARE-Attn

A.1 Tile-wise $\boldsymbol{QK}$ Residual Measurement

Measurement. The maps and ranking studies in Figures 3, 7, and 5 isolate the $\boldsymbol{QK}$ pathway by fake-quantizing only $\boldsymbol{Q}$ and $\boldsymbol{K}$ with NVFP4, keeping $\widetilde{\boldsymbol{P}}$ and $\boldsymbol{V}$ in high precision. On each query row, we measure the total variation $\mathrm{TV}_i \in [0, 1]$ in equation 4 between the two attention distributions, using a shared row-max to form both softmax numerators. Unlike output-space error, this TV splits exactly across key tiles: a tile's color in Figure 5 is the probability mass it moves onto the wrong keys; we refer to this quantity as TV or misplaced probability mass throughout. Measurements use Wan2.1-14B after 400 steps of Attn-QAT, sequence length $L = 76{,}800$ (81 frames at 720×1280), 40 layers $\times$ 40 heads, and 128×128 FlashAttention tiles (600 key tiles per query row). Whenever a predictor or allocation table is fit, the calibration and evaluation data come from disjoint prompts.

Residual maps. Figure 5 shows the equally spaced (layer, head) grid at four points along the sampling trajectory. The sparse heads remain sparse, and the diffuse heads remain diffuse: the (layer, head) pattern is stable, which is why we allocate the correction budget per (layer, head) and do not allocate per denoising step (Figure 7d). Figure 6 further shows that QAT's layer-wise reduction and a 1% sparse budget are not substitutes: QAT acts mainly on late layers, whereas sparse recovery is spread across depth, including the mid-depth error peak that QAT barely reduces.



(a) Step 5.

(b) Step 17.

(c) Step 29.

(d) Step 41.

Figure 5: **Residual QK error is sparse across heads and stable across denoising steps.** Tile-wise total variation between high-precision and QK-quantized attention probabilities on Wan2.1-14B after Attn-QAT. Each panel is one (layer, head); rows are sampled query tiles and columns are 128-token key tiles. Color is the tile's contribution to the TV (log scale; values below 0.02% in white). The number is the share of TV in the top 1% of key tiles. Sparsity is a (layer, head) property: at step 29 and layer 33, the top 1% accounts for 8% in head 0 but 94% in head 39. The same heads stay sparse or diffuse from step 5 through 41.

A.2 Adaptive Sparse QK Correction

Offline budget allocation. For each unit $u = (\ell, h)$, we measure a discrete TV recovery curve

$$g_u(b) = \mathrm{tv}_u^0 - \mathrm{tv}_u(b),$$

where tv_u^0 is the residual TV without correction and $\mathrm{tv}_u(b)$ is the remaining TV under correction budget b. We evaluate each curve at nominal budgets of 0.25%, 0.5%, 1%, 2%, 5%, and 10%, each rounded to a whole number of the 600 key tiles, so that the optimized budget matches the actual kernel cost.

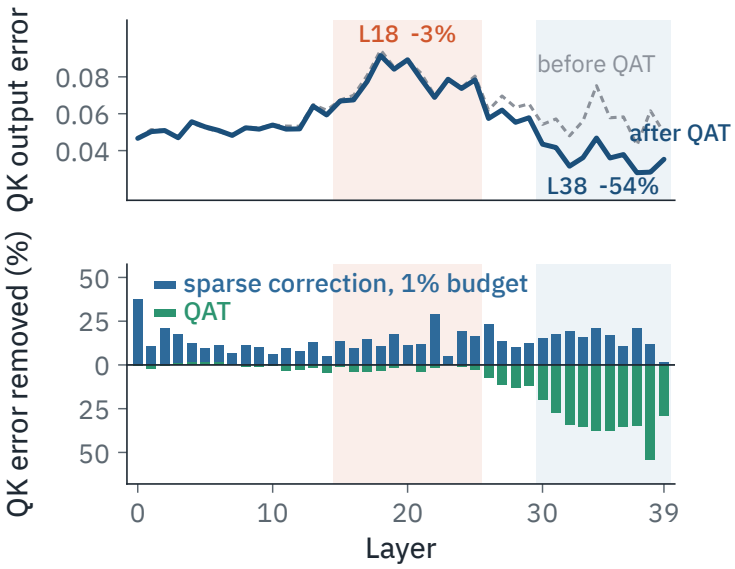


Figure 6: **QAT and a 1% sparse budget act on different layers.** Wan2.1-14B, same Attn-QAT checkpoint as Figure 1. Top: per-layer $e_{\boldsymbol{QK}}$ before and after QAT. QAT barely moves the mid-depth error peak (layer 18: -3%) and concentrates its reduction in late layers (layer 38: -54%). Bottom: $\boldsymbol{QK}$ error removed by Attn-QAT versus by a 1% high-precision tile budget. Sparse recovery is spread across depth, and QAT's reduction opens only after layer 26.

We replace each empirical recovery curve by its upper concave envelope and solve

$$\max_{\{b_u\}} \sum_u g_u(b_u) \qquad \text{s.t.} \qquad \frac{1}{|\mathcal{U}|} \sum_u b_u \le B. \tag{15}$$

With Lagrange multiplier λ, each unit independently chooses

$$b_u(\lambda) = \arg\max_b \left[g_u(b) - \lambda b \right], \tag{16}$$

and λ is found by binary search. The resulting table is calibrated once and frozen. For the 1% setting, the realized average budget over the 1600 layer–head pairs is 0.9984%.

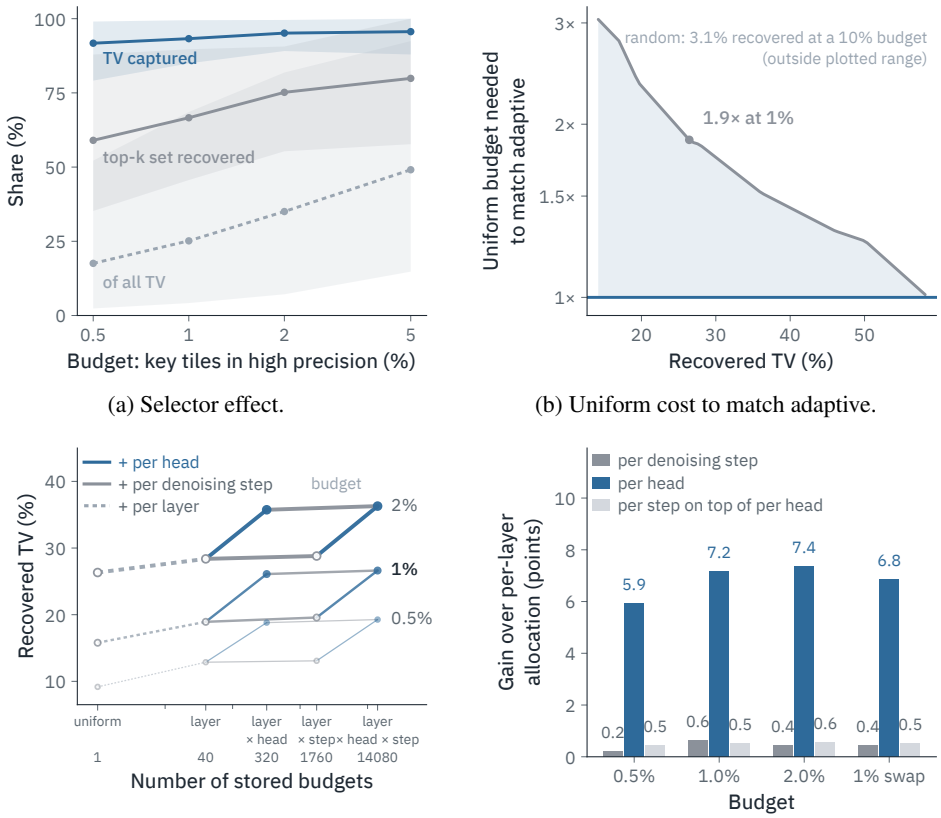
We use (layer, head) as the allocation unit because head-wise heterogeneity is substantially larger than variation across denoising steps. On a held-out prompt, per-(layer, head) allocation raises TV recovery at a 1% average budget from 17.0% under a uniform budget to 26.7%; Appendix A.3 compares it with per-layer and per-step allocation.

Online tile selection. For every query tile, we independently rank key tiles using the mean-pooled score in equation 7. The predictor closely tracks the true tile-wise TV contribution. Its mean Spearman correlation is 0.886 across the evaluated layer–head pairs. On a separate grid of 5 denoising steps and 8 layers (head 0), its mean Spearman correlation with the tile-wise TV (0.85) is close to that of a selector that inspects the full FP4 logits (0.94). Heads with different budgets in the same layer share one top-k up to the largest budget, and each head keeps the prefix of its own ranking.

Forward–backward consistency. The same correction mask is used in the forward pass and in both backward logit-recomputation paths. Otherwise, the backward pass would recompute uncorrected logits and differentiate a softmax distribution different from the one used in the forward pass.

A.3 Selector Recovery, Budget Cost, and Allocation Granularity

Figure 3c in the main text shows recovered TV versus the high-precision key-tile budget. Figure 7 reports how well the mean-pool selector hits the true high-error tiles, the cost of a uniform budget, and the choice of table granularity. Beyond per-layer allocation, nearly all of the additional gain



(a) Selector effect.

(b) Uniform cost to match adaptive.

(c) Allocation granularity.

(d) Per-head versus per-step allocation.

Figure 7: **Selector recovery, uniform-budget cost, and allocation granularity.** All curves are scored on a held-out prompt. (a) At a 1% key-tile budget the selector captures 93.3% of the TV carried by the true top-1% tiles (set overlap 66.6%). (b) Budget that uniform allocation needs to match adaptive recovery; $1.9\times$ at a 1% budget. Random selection recovers only 3.1% at a 10% budget, outside the plotted range. (c) Recovery versus the number of stored budgets at $0.5\%/1\%/2\%$ average budgets (line width). Edge color marks the dimension along which the allocation is refined (steel-blue: head; gray: denoising step; dashed: layer). Axis labels give the allocation scheme and its number of stored budgets. (d) Gain over per-layer allocation from refining per head ($+280$ budgets) versus per denoising step ($+1{,}720$ budgets), including a run with the calibration and held-out prompts swapped.

comes from refining the table per head rather than per denoising step. The granularity comparison uses a subset of 8 heads per layer; the deployed table covers all 40 heads.

A.4 $\boldsymbol{P}$-EXACT QAT

Two-level probability scaling. The softmax numerator $\widetilde{P}_{ij} = \exp(S_{ij} - m_i)$ is concentrated near zero. In a FlashAttention kernel, it is formed tile by tile with a running row maximum m_i, so within a key tile J its row maximum $\max_{j \in J} \widetilde{P}_{ij} \leq 1$ can be far below one. Direct NVFP4 microscaling therefore underuses the E4M3 scale range and can severely quantize or underflow low-magnitude probability blocks. Following SageAttention3 (Zhang et al., 2025b), we first apply a high-precision row-level scale within each key tile and then native NVFP4 microscaling. For row i and key tile J,

$$s^{(1)}_{\boldsymbol{P},iJ} = \frac{\max_{j \in J} \widetilde{P}_{ij}}{448 \times 6}, \qquad \widetilde{\boldsymbol{P}}^{(1)}_{iJ} = \frac{\widetilde{\boldsymbol{P}}_{iJ}}{s^{(1)}_{\boldsymbol{P},iJ}}, \tag{17}$$

where 448 and 6 are the maxima of the E4M3 scale and E2M1 value. The standard NVFP4 quantizer $\mathcal{Q}$ is then applied to $\widetilde{\boldsymbol{P}}^{(1)}$:

$$\mathcal{Q}\big(\widetilde{\boldsymbol{P}}^{(1)}_{in}\big) = s^{(2)}_{\boldsymbol{P},in}\,\widehat{\boldsymbol{P}}_{in}, \tag{18}$$

where n indexes 16-element micro-blocks within the tile, $s^{(2)}_{\boldsymbol{P},in}$ is the E4M3 micro-block scale, and $\widehat{\boldsymbol{P}}_{in}$ holds the E2M1 values. The numerator is reconstructed as

$$\widetilde{\boldsymbol{P}}_{in} \approx s^{(1)}_{\boldsymbol{P},iJ}\, s^{(2)}_{\boldsymbol{P},in}\,\widehat{\boldsymbol{P}}_{in}, \tag{19}$$

which is the map $\mathcal{Q}^{\mathrm{2L}}_{\boldsymbol{P}}(\widetilde{\boldsymbol{P}})$ used at inference in Section 4.2. The first level stretches each row of the tile onto the full E4M3 range; the second retains fine-grained NVFP4 scaling.

Backward computation under $\boldsymbol{P}$-exact QAT. During $\boldsymbol{P}$-exact QAT, the softmax numerator is not fake-quantized:

$$\boldsymbol{O}^{\mathrm{train}}_{i,:} = \frac{1}{\widehat{\ell}_i}\sum_j \widehat{\widetilde{P}}_{ij}\boldsymbol{V}^F_{j,:}. \tag{20}$$

The same unquantized probability weights are used to compute

$$d\boldsymbol{V} = \boldsymbol{P}^\top d\boldsymbol{O}, \tag{21}$$

rather than using a fake-quantized probability operand. This removes the probability-induced perturbation in the value-gradient path. It also eliminates the auxiliary output $\boldsymbol{O}'$ that Attn-QAT accumulates from unquantized probabilities for the FlashAttention backward pass: under $\boldsymbol{P}$-exact QAT, $\boldsymbol{O}^{\mathrm{train}}$ is exactly this output, so the additional high-precision $\widetilde{\boldsymbol{P}}\boldsymbol{V}$ accumulation and its buffer are unnecessary.

At inference, we restore the deployment quantizer $\mathcal{Q}^{\mathrm{2L}}_{\boldsymbol{P}}$. Thus, $\boldsymbol{P}$-exact QAT deliberately decouples training and inference precision: probability quantization is omitted during optimization but retained in the deployed FP4 attention kernel.

A.5 Mean-Centered $\boldsymbol{V}$ Quantization

Let $A_{ij} = \widehat{\widetilde{P}}_{ij}/\widehat{\ell}_i$ and $\boldsymbol{c} = N_k^{-1}\sum_m \boldsymbol{V}_{m,:}$. The high-precision mean-centered output is $O_{i,:} = \sum_j A_{ij}(\boldsymbol{V}_{j,:} - \boldsymbol{c}) + \boldsymbol{c}$. For channel d,

$$\frac{\partial O_{i,d}}{\partial V_{m,d}} = A_{im} - \frac{1}{N_k}\sum_j A_{ij} + \frac{1}{N_k} = A_{im}, \tag{22}$$

since $\sum_j A_{ij} = 1$. Unlike scaling or rotation, mean-centering therefore adds no extra high-precision gradient correction.

The inference-time compensation with quantized probability mass α_i is given in equation 14. If $\boldsymbol{V}$ were not quantized, equation 14 would reduce exactly to attention over the original $\boldsymbol{V}$ with the same quantized probabilities.

A.6 Residual Structure in Language Models

Table 4 repeats the component-wise analysis of Section 3 on Qwen3-8B and Llama-3.1-8B, using held-out C4 text at sequence length 8192. After Attn-QAT, $e_{\boldsymbol{QK}}$ remains the largest term, 2.3–$5.2\times$ the two-level $e_{\boldsymbol{P}}$ and $e_{\boldsymbol{V}}$, which motivates $\boldsymbol{QK}$ correction. Two-level scaling reduces $e_{\boldsymbol{P}}$ by 2.1–$5.4\times$, so the $\boldsymbol{P}$ error is again quantizer-limited. The $\boldsymbol{V}$ residual has a different structure from diffusion transformers. It is dominated by the attention sink (Xiao et al., 2024): the first token receives a large share of the attention and has a large value norm, and because NVFP4 blocks for $\boldsymbol{V}$ run along the token axis, it acts as a token-wise outlier rather than a channel-wise offset. Mean-centering therefore does not help; on the pretrained Qwen3-8B weights, it raises $e_{\boldsymbol{V}}$ from 0.037 to 0.062. For $\boldsymbol{QK}$ budget allocation, the perplexity gain of correction is strongly front-loaded: on Qwen3-8B, the first corrected tile per query tile recovers 60.5% and 71.3% of the perplexity gain of computing all $\boldsymbol{QK}$ tiles in high precision on Attn-QAT and CARE-Attn weights, respectively. A uniform budget therefore gives every head its most valuable tiles.

Table 4: **Component-wise attention-output error in language models** (held-out C4, length 8192). $e^1_{\boldsymbol{P}}$ and $e^2_{\boldsymbol{P}}$: single- and two-level $\boldsymbol{P}$ quantization.

Model	Weights	$e_{\boldsymbol{QK}}$	$e^1_{\boldsymbol{P}}$	$e^2_{\boldsymbol{P}}$	$e_{\boldsymbol{V}}$	e_{all}
Qwen3-8B	pretrained	0.1548	0.0776	0.0284	0.0372	0.1660
Qwen3-8B	Attn-QAT	0.1592	0.0642	0.0306	0.0432	0.1736
Llama-3.1-8B	pretrained	0.1298	0.2137	0.0399	0.0607	0.1560
Llama-3.1-8B	Attn-QAT	0.1002	0.1348	0.0373	0.0433	0.1234

B Experimental Details

B.1 Training and Evaluation Setup

Video. Table 5 lists the training and sampling settings, which follow Attn-QAT. On Wan2.1-14B, the Attn-QAT baseline is the checkpoint released by its authors[1], trained with this recipe. All other 14B runs are trained by us on 128 GPUs and differ only in the attention path; the BF16 fine-tune uses the same recipe with FlashAttention. We additionally reproduce Attn-QAT with the same recipe, GPU count, and batch configuration (Appendix B.3). The training data are synthetic Wan-generated videos from FastVideo: the 77-frame 768×1280 set[2] for Wan2.1-14B, and an 81k-video subset of the 77-frame 448×832 set[3] for Wan2.1-1.3B. Within each model, all runs share the prompts and the generation seed, so scores are paired per prompt. Wan2.1-14B uses all 326 VBench prompts with LLM-augmented text; Wan2.1-1.3B uses the original short text of VBench.

Fidelity. For each prompt, we compare the generated video frame by frame with the video generated by the BF16 fine-tuned model from the same prompt and seed; training-free methods are compared with the original BF16 model instead. We use all 81 frames at 1280×720. PSNR and SSIM are computed on RGB frames and averaged over frames, and LPIPS uses the AlexNet backbone. We report means over the 326 prompts and, for paired comparisons, the fraction of prompts on which LPIPS decreases.

Training cost. Step times are median per-iteration wall-clock times after step 20. On Wan2.1-14B, single-level fake-quantized $\boldsymbol{P}$ (Attn-QAT), two-level fake-quantized $\boldsymbol{P}$, and $\boldsymbol{P}$-exact QAT take 88.35, 93.36, and 72.77 s per step; mean-centered $\boldsymbol{V}$ adds 0.6 s, and a uniform 1% $\boldsymbol{QK}$ budget another 10.0 s, so a training step with all three corrections (83.37 s) remains faster than Attn-QAT.

Table 5: **Video training and sampling settings.**

	Wan2.1-14B	Wan2.1-1.3B
Training data	Wan-Syn, 768×1280	Wan-Syn, 448×832 (81k subset)
GPUs	128	16
Global batch / grad. accum.	32 / 1	16 / 1
Optimizer	AdamW (β=(0.9, 0.999), wd 0.01), constant lr 1×10^{-6}, no warmup	
Steps / seed	400 / 1000	4000 / 1000
Resolution, frames, fps	1280×720, 81, 16	832×480, 81, 16
Sampling steps / guidance	50 / 5.0	50 / 3.0
Sampling flow shift	5.0	3.0

Language models. Training follows Attn-QAT (Zhang et al., 2026b). We use AdamW with learning rate 5×10^{-6}, sequence length 8192, global batch size 16, and 2000 steps (262M tokens), with BF16 mixed precision and activation checkpointing. Qwen3-32B uses a global batch size of 32 for 1000 steps, which keeps the same token budget; its BF16 fine-tune was not run due to limited resources. Training data are the first two English C4 training shards, re-packed with each model's

[1] https://huggingface.co/FastVideo/14B_qat_400

[2] https://huggingface.co/datasets/FastVideo/Wan-Syn_77x768x1280_250k

[3] https://huggingface.co/datasets/FastVideo/Wan-Syn_77x448x832_600k

tokenizer into 33,000 sequences of length 8192. Each run uses one node with 8 RTX PRO 5000 GPUs and gradient accumulation 2, except Qwen3-32B, which uses 32 GPUs (4 nodes) with fully sharded data parallelism and no gradient accumulation. C4 perplexity is computed on the first 64 held-out sequences of length 8192. WikiText-2 perplexity is computed on the full test set, packed into sequences of length 8192. CARE-Attn recomputes three key tiles per query tile in high precision during both training and inference. $\boldsymbol{V}$ is not centered: its channel mean over all value tokens would depend on future tokens and would change as the KV cache grows during decoding, requiring the cached values to be requantized.

B.2 Additional Language-Model Results

Results per run. Table 6 lists C4 and WikiText-2 perplexity for every run; Table 2 averages the two evaluation sets and, for the 8B models, the two seeds. CARE-Attn improves over Attn-QAT by 0.06–0.26 on C4 and 0.05–0.18 on WikiText-2, and it is below Attn-QAT + $\boldsymbol{QK}$ corr. in every run and on both evaluation sets. Where a BF16 reference is available (all models except Qwen3-32B), it closes 71%–87% of the gap between Attn-QAT and BF16 on C4 and 64%–80% on WikiText-2.

Table 6: **Perplexity of every language-model run** at length 8192 (lower is better). Exact $\boldsymbol{QK}$: CARE-Attn weights with every $\boldsymbol{QK}$ tile in high precision. BF16 FT: BF16 fine-tune. Gap closed $= 1 - (\text{CARE-Attn} - \text{BF16})/(\text{Attn-QAT} - \text{BF16})$. Bold marks the best FP4 method.

Model	Seed	Attn-QAT	+ $\boldsymbol{QK}$ corr.	CARE-Attn	Exact $\boldsymbol{QK}$	BF16 FT	Gap closed
C4							
Qwen3-8B	42	11.5407	11.3931	**11.3185**	11.2854	11.2363	73.0%
Qwen3-8B	7	11.5411	11.3939	**11.3178**	11.2833	11.2363	73.3%
Llama-3.1-8B	42	9.5480	9.3589	**9.2932**	9.2765	9.2347	81.3%
Llama-3.1-8B	7	9.5555	9.3685	**9.2908**	9.2756	9.2347	82.5%
Mistral-7B-v0.3	42	7.9231	7.8664	**7.8620**	7.8537	7.8528	86.9%
Qwen3-14B	42	10.5989	10.4943	**10.4480**	10.4224	10.3876	71.4%
Qwen3-32B	42	10.3699	10.2923	**10.2635**	10.2428	–	–
WikiText-2							
Qwen3-8B	42	6.5404	6.4477	**6.4199**	6.3854	6.3530	64.3%
Qwen3-8B	7	6.5621	6.4676	**6.4210**	6.3874	6.3530	67.5%
Llama-3.1-8B	42	6.0002	5.8846	**5.8193**	5.7895	5.7737	79.9%
Llama-3.1-8B	7	5.9906	5.8751	**5.8199**	5.7920	5.7737	78.7%
Mistral-7B-v0.3	42	5.2400	5.1977	**5.1800**	5.1659	5.1575	72.7%
Qwen3-14B	42	5.9001	5.8358	**5.8195**	5.7931	5.7761	65.0%
Qwen3-32B	42	5.8242	5.7802	**5.7714**	5.7541	–	–

Training-side ablation. Table 7 separates the training-side changes of CARE-Attn on Qwen3-8B; every entry uses 3-tile $\boldsymbol{QK}$ correction at inference. The $\boldsymbol{P}$ correction has a training half ($\boldsymbol{P}$-exact QAT) and an inference half (two-level $\boldsymbol{P}$), and the inference half helps only when used together with the training half. Two-level $\boldsymbol{P}$ alone raises the C4 perplexity of Attn-QAT by 0.029, and $\boldsymbol{P}$-exact QAT alone lowers it by 0.019, whereas together they lower it by 0.053. Training with $\boldsymbol{QK}$ correction lowers it by 0.018, and all changes together lower it by 0.075.

Table 7: **Training-side ablation on Qwen3-8B** (seed 42; C4 PPL at 8192). Columns give the $\boldsymbol{P}$ quantizer at inference; all entries use 3-tile $\boldsymbol{QK}$ correction at inference.

Training	Single-level $\boldsymbol{P}$	Two-level $\boldsymbol{P}$
Attn-QAT	11.3931	11.4216
+ $\boldsymbol{P}$-exact QAT	11.3744	11.3401
+ $\boldsymbol{QK}$ corr. in training	11.3747	–
+ both (CARE-Attn)	11.3422	**11.3185**

$\boldsymbol{QK}$ correction budget. Table 8 varies the number of corrected key tiles per query tile on Qwen3-8B at inference. Perplexity decreases with the budget for both weights, and CARE-Attn weights are 0.07–0.08 below Attn-QAT weights at every budget. The main results use three tiles.

Table 8: $\boldsymbol{QK}$ **correction budget on Qwen3-8B** (seed 42; C4 PPL at 8192). Under causal masking, the fraction of reachable key tiles that is recomputed is about twice the listed fraction.

Key tiles per query tile	1	2	3	4	6	8
Fraction of key tiles	1.56%	3.13%	4.69%	6.25%	9.38%	12.50%
Attn-QAT weights	11.4268	11.4037	11.3931	11.3864	11.3771	11.3717
CARE-Attn weights	**11.3535**	**11.3260**	**11.3185**	**11.3119**	**11.3032**	**11.2996**

Longer contexts. Table 9 evaluates the length-8192 checkpoints at 16k and 32k tokens. CARE-Attn has the lowest perplexity in all eight settings, with the same three corrected key tiles per query tile at every length.

Table 9: **C4 perplexity at longer contexts** (seed 42; models trained at length 8192).

Model	Length	Attn-QAT	+ $\boldsymbol{QK}$ corr.	CARE-Attn
Qwen3-8B	8k	11.5407	11.3931	**11.3185**
Qwen3-8B	16k	11.4517	11.2783	**11.2031**
Qwen3-8B	32k	10.8715	10.6851	**10.6783**
Llama-3.1-8B	8k	9.5480	9.3589	**9.2932**
Llama-3.1-8B	16k	9.8083	9.6095	**9.5478**
Llama-3.1-8B	32k	9.2498	9.0279	**8.9275**
Mistral-7B-v0.3	8k	7.9231	7.8664	**7.8620**
Mistral-7B-v0.3	32k	7.7398	7.6658	**7.6485**

Long-context retrieval. Table 10 evaluates synthetic needle-in-a-haystack retrieval in the style of RULER (Hsieh et al., 2024) on the length-8192 checkpoints, with haystacks drawn from held-out C4 text. In the 16-needle task, 16 needles with single-word keys are inserted, and one of them is queried; in the UUID task, 8 needles with UUID keys and values are inserted, following the `niah_multikey_3` setting of RULER. We report exact match under greedy decoding, computed from a single teacher-forced forward pass: an answer is correct if every answer token is the argmax. All methods answer the same 400 questions per setting (200 at 64k). Both CARE-Attn and Attn-QAT + $\boldsymbol{QK}$ corr. correct three key tiles per query tile at every length, so the fraction of causally reachable key tiles recomputed in high precision falls from 4.6% at 16k to 2.3% at 32k and 1.2% at 64k. CARE-Attn is more accurate than both baselines in every setting, and the gap widens with length: on Llama-3.1-8B UUID, its lead over Attn-QAT grows from 13.0 points at 16k to 20.5 at 32k and 46.0 at 64k, whereas adding $\boldsymbol{QK}$ correction to Attn-QAT only at inference improves it by 6.0 points at 64k.

Table 10: **Long-context retrieval** (exact match, %; higher is better; seed 42; models trained at length 8192; three corrected key tiles per query tile). Bold marks the best FP4 method.

Model	Length	Task	Attn-QAT	+ $\boldsymbol{QK}$ corr.	CARE-Attn	BF16 fine-tune
Qwen3-8B	16k	16-needle	74.0	81.2	**83.0**	87.8
Qwen3-8B	16k	UUID	73.2	80.8	**84.0**	84.8
Qwen3-8B	32k	16-needle	57.0	60.0	**73.2**	83.2
Qwen3-8B	32k	UUID	59.0	67.0	**77.8**	80.2
Llama-3.1-8B	16k	16-needle	79.2	82.0	**89.5**	94.2
Llama-3.1-8B	16k	UUID	67.8	69.0	**80.8**	88.5
Llama-3.1-8B	32k	16-needle	60.2	74.0	**80.0**	91.8
Llama-3.1-8B	32k	UUID	51.0	58.2	**71.5**	82.2
Llama-3.1-8B	64k	16-needle	40.5	41.0	**64.0**	81.5
Llama-3.1-8B	64k	UUID	20.5	26.5	**66.5**	84.0

B.3 Additional Diffusion Results

Robustness to the Attn-QAT run. Besides the released checkpoint, we reproduce Attn-QAT with the same recipe, GPU count, and batch configuration as CARE-Attn (Table 11). The two runs differ by 0.0123 in imaging quality and 0.0045 in weighted score. CARE-Attn exceeds both in imaging quality, aesthetic quality, and weighted score. Compared with the stronger of the two, it improves imaging quality by 0.0075, aesthetic quality by 0.0067, and the weighted score by 0.0037.

Table 11: **Two Attn-QAT runs on Wan2.1-14B** (326 prompts; higher is better). Both runs use the recipe of Table 5. Bold marks the best FP4 method.

Method	Imaging	Aesthetic	Subject	Background	Flicker	Smoothness	Dynamic	Weighted
Attn-QAT (released)	0.6823	0.6680	**0.9671**	**0.9700**	**0.9809**	**0.9888**	0.3926	0.8390
Attn-QAT (reproduced)	0.6946	0.6693	0.9637	0.9683	0.9781	0.9880	0.4417	0.8435
CARE-Attn	**0.7021**	**0.6760**	0.9654	0.9687	0.9768	0.9879	**0.4601**	**0.8472**

Fidelity to the BF16 fine-tune. Table 12 lists PSNR, SSIM, and LPIPS against the BF16 fine-tune. CARE-Attn is 4.42 and 3.16 dB closer than the released and reproduced Attn-QAT, with lower LPIPS on 99% and 97% of the prompts. Adding the inference-side corrections to Attn-QAT weights does not close this gap: on the reproduced run, PSNR drops by 0.83 dB. With the same inference-side corrections, weights trained with them (CARE-Attn with a uniform 1% budget, 15.47 dB) are 3.69 dB closer than the reproduced Attn-QAT weights with the same corrections, with lower LPIPS on 99% of the prompts. Within CARE-Attn, mean-centered $\boldsymbol{V}$ and $\boldsymbol{QK}$ correction together add 1.2 dB over $\boldsymbol{P}$-exact QAT, with lower LPIPS on 79% of the prompts.

Table 12: **Fidelity to the BF16 fine-tune on Wan2.1-14B** (326 prompts, same prompt and seed). + corr.: two-level $\boldsymbol{P}$, mean-centered $\boldsymbol{V}$, and a uniform 1% $\boldsymbol{QK}$ budget at inference, without retraining. Bold marks the best method.

Method	PSNR↑	SSIM↑	LPIPS↓
BF16 (original)	10.98	0.4079	0.5893
Attn-QAT (released)	11.34	0.4448	0.5434
Attn-QAT (reproduced)	12.60	0.4913	0.4807
Attn-QAT (reproduced) + corr.	11.78	0.4333	0.5424
$\boldsymbol{P}$-exact QAT	14.53	0.5726	0.3690
+ $\boldsymbol{V}$ centering	15.10	0.5890	0.3475
+ $\boldsymbol{QK}$ corr. (CARE-Attn)	**15.76**	**0.6162**	**0.3147**

Corrections without training. Table 13 applies the inference-side corrections to the original Wan2.1-14B weights without any training; fidelity is measured against the original BF16 model. Two-level $\boldsymbol{P}$, mean-centered $\boldsymbol{V}$, and $\boldsymbol{QK}$ correction together raise the weighted score from 0.8127 to 0.8392, reaching that of Attn-QAT after 400 training steps (0.8390). At the same 4-bit width, they are 2.67 dB closer to the BF16 output than SageAttention3, with lower LPIPS on 93% of the prompts. This setting uses a uniform 10% $\boldsymbol{QK}$ budget (about $1.40\times$ attention cost), so its cost is not comparable to the 1% budget used elsewhere.

Table 13: **Inference-side corrections on the original Wan2.1-14B** (326 prompts). VBench dimensions and PSNR/SSIM: higher is better; LPIPS: lower is better. Fidelity is measured against the original BF16 model.

Method	Imaging	Aesthetic	Subject	Background	Flicker	Smoothness	Dynamic	Weighted	PSNR	SSIM	LPIPS
SageAttention3	0.6723	0.6517	0.9528	0.9584	0.9756	0.9859	0.4509	0.8342	15.33	0.5289	0.4346
FP4	0.6457	0.6214	0.9442	0.9545	0.9780	0.9853	0.3067	0.8127	13.49	0.4548	0.5260
+ two-level $\boldsymbol{P}$	0.6472	0.6367	0.9517	0.9568	0.9784	0.9861	0.2699	0.8141	14.88	0.5044	0.4752
+ mean-centered $\boldsymbol{V}$ + 10% $\boldsymbol{QK}$ corr.	0.6931	0.6679	0.9519	0.9558	0.9740	0.9863	0.4509	0.8392	18.00	0.6142	0.3212

Wan2.1-1.3B. Table 14 reports VBench on Wan2.1-1.3B after 4000 training steps. Attn-QAT is served as on Wan2.1-14B, and CARE-Attn uses a uniform 1% $\boldsymbol{QK}$ budget. CARE-Attn raises the weighted score from 0.8283 to 0.8317, mainly through aesthetic quality ($+0.0204$) and imaging quality ($+0.0066$), and closes nearly half of the gap to the BF16 fine-tune (0.8356). Mean-centered $\boldsymbol{V}$ contributes to the aesthetic gain: over an otherwise identical run trained without it, aesthetic quality improves by 0.0149.

Table 14: **VBench results on Wan2.1-1.3B** (step 4000; higher is better). Bold marks the best FP4 method.

Method	Imaging	Aesthetic	Subject	Background	Flicker	Smoothness	Dynamic	Weighted
BF16 fine-tune	0.6709	0.6245	0.9671	0.9683	0.9790	0.9867	0.4693	0.8356
Attn-QAT	0.6739	0.6011	0.9665	0.9680	0.9766	0.9851	**0.4260**	0.8283
CARE-Attn	**0.6805**	**0.6215**	**0.9698**	**0.9682**	**0.9790**	**0.9870**	0.4007	**0.8317**

B.4 Visual Comparison

Figures 8–11 show videos generated by Wan2.1-14B from the same prompt and seed. In each block, the rows are the BF16 fine-tune, Attn-QAT, and CARE-Attn, and the columns are frames 0, 40, and 80. The Attn-QAT row is the released checkpoint in Figures 8 and 9 and the reproduced run in Figures 10 and 11, although both are labeled Attn-QAT in the figures. Attn-QAT tends to darken the scene, lower contrast, and blur details, while CARE-Attn stays close to the BF16 fine-tune. Over all 326 prompts, CARE-Attn has higher imaging quality than the released Attn-QAT on 62.6% of the prompts and higher aesthetic quality on 57.1%.

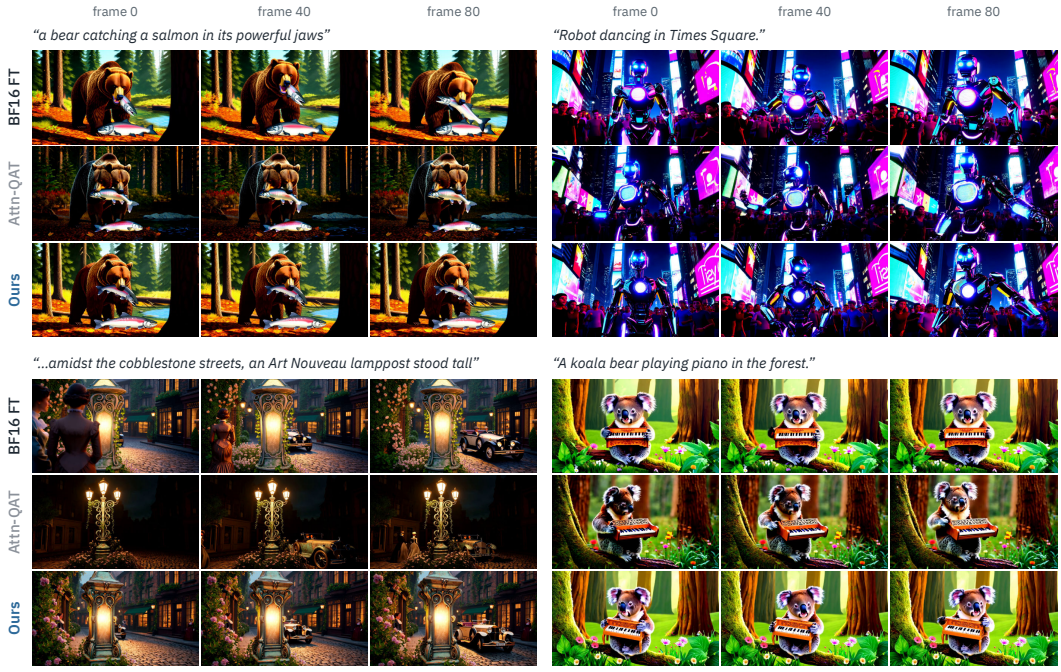


Figure 8: **Visual comparison with the released Attn-QAT on Wan2.1-14B.** Rows: BF16 fine-tune, Attn-QAT, and CARE-Attn; columns: frames 0, 40, and 80.

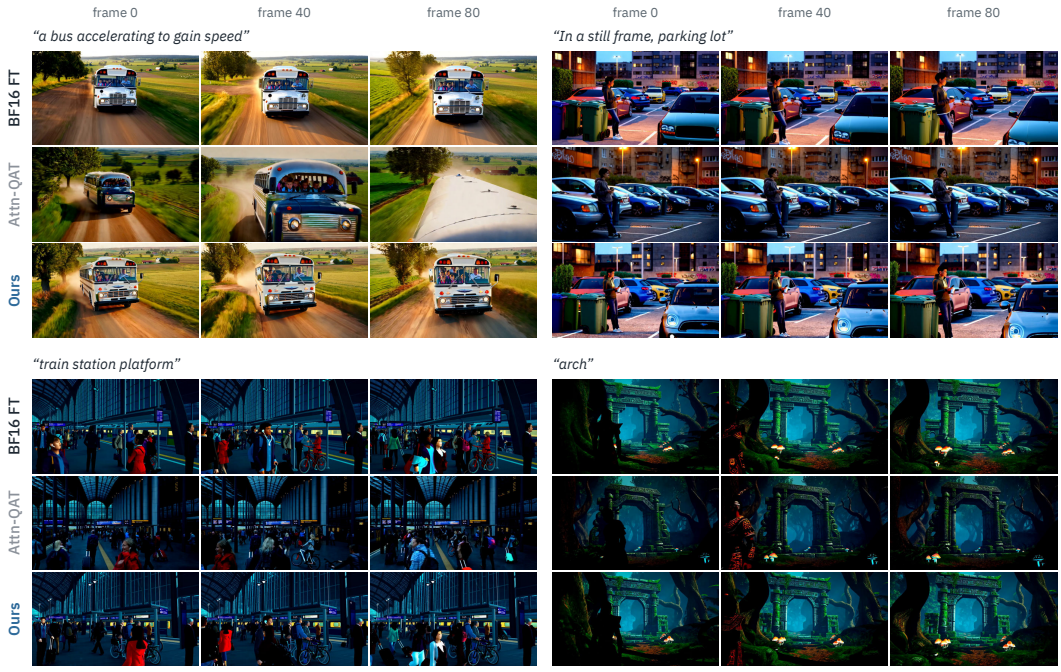


Figure 9: **Visual comparison with the released Attn-QAT on Wan2.1-14B.** Rows: BF16 fine-tune, Attn-QAT, and CARE-Attn; columns: frames 0, 40, and 80.

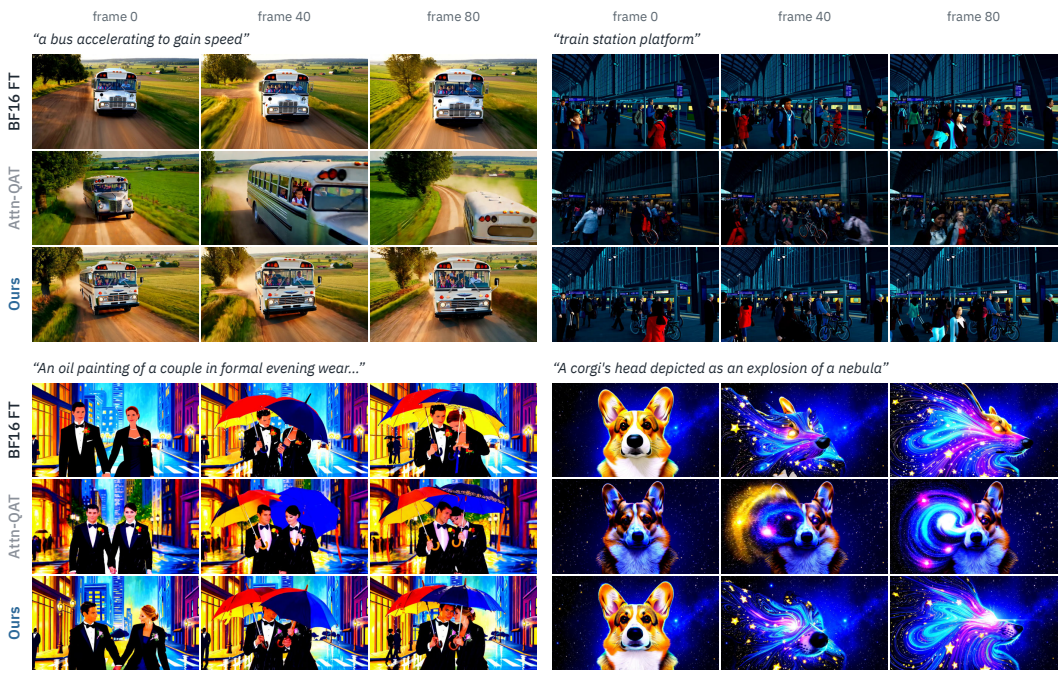


Figure 10: **Visual comparison with the reproduced Attn-QAT on Wan2.1-14B.** Rows: BF16 fine-tune, Attn-QAT, and CARE-Attn; columns: frames 0, 40, and 80. The row labeled Attn-QAT is the reproduced run.

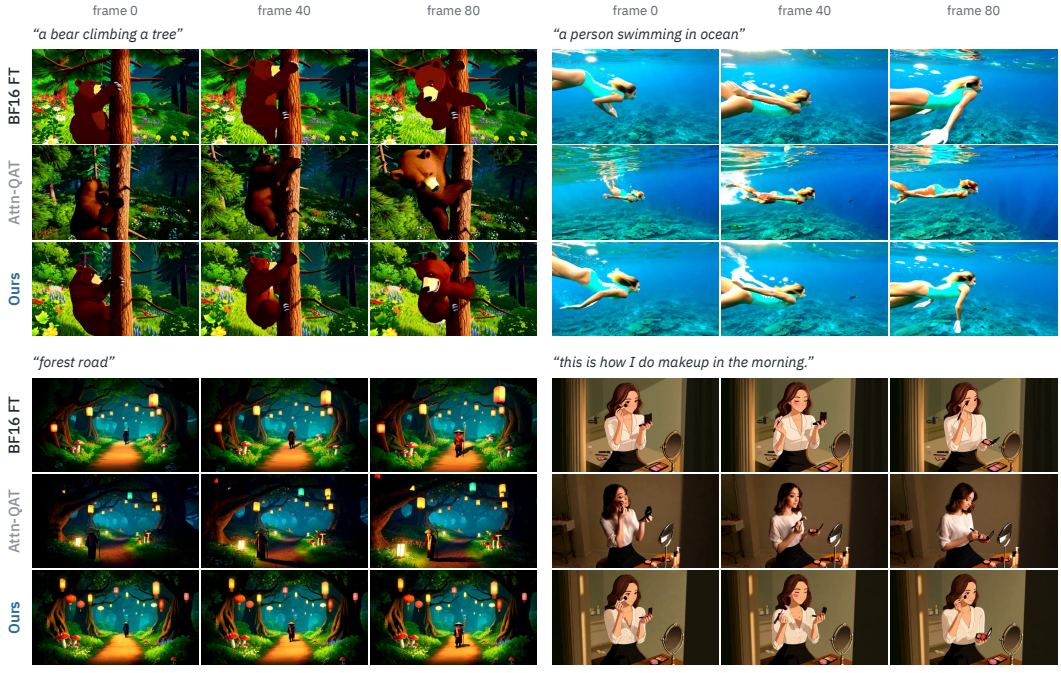


Figure 11: **Visual comparison with the reproduced Attn-QAT on Wan2.1-14B.** Rows: BF16 fine-tune, Attn-QAT, and CARE-Attn; columns: frames 0, 40, and 80. The row labeled Attn-QAT is the reproduced run.