ROTATEATTENTION-FP4: A UNIFIED TRAINING-FREE AND DATA-EFFICIENT QAT FRAMEWORK FOR QUANTIZED ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Attention incurs substantial computation and memory traffic in video diffusion models, motivating 4-bit quantization for efficient inference. However, existing training-free 4-bit attention methods remain limited in both generation quality and runtime efficiency, while quantization-aware training (QAT) often incurs substantial training costs. We propose **RotateAttention-FP4**, a unified NVFP4 attention framework supporting accurate and efficient video diffusion through both training-free inference and lightweight adaptation. Our key insight is that activation outliers and stable channel-wise biases amplify quantization error, degrading the accuracy of training-free inference and increasing the burden on model adaptation. We refer to these distribution-induced errors as *structural errors* and suppress them through grouped Hadamard rotation and calibration-based channel centering using offline-estimated offsets. We further optimize quantization scales for both QKV and attention weights: rotation-aware QKV scale selection reduces FP4 rounding error, while probability amplification reduces underflow of attention-weight scales. Together, these corrections improve training-free inference accuracy and reduce the residual error that sensitive models must compensate for through lightweight QAT. Experiments on Wan2.2 and MiniMax H3 demonstrate generation quality close to 16-bit attention **without any fine-tuning** in conventional multi-step settings and on eight-step MiniMax H3. For the quantization-sensitive four-step Wan2.2 model, QAT with low-rank adaptation (LoRA) recovers generation quality using **only 30 randomly sampled training examples**. Our fused implementation further delivers a 2.11–$2.33\times$ **end-to-end attention speedup over FlashAttention** and 1.18–$1.41\times$ **preprocessing throughput over SageAttention3**.

1 INTRODUCTION

Recent diffusion models have achieved remarkable progress in high-fidelity video generation, but their inference cost grows rapidly with model size, spatial-temporal resolution, and the number of denoising steps. Among different components, attention introduces substantial computation and memory traffic, making **4-bit attention quantization** a promising direction for efficient video generation.

Existing low-bit attention methods mainly rely on post-training quantization (PTQ) (Zhang et al., 2025a; 2026a) or quantization-aware training (QAT) (Zhang et al., 2026b). PTQ is training-free and easy to deploy, but can suffer severe quality degradation on quantization-sensitive models at 4-bit precision (Zhang et al., 2026b). Diffusion models distilled to only one or a few denoising steps pose this challenge: prior work reports substantial quality degradation after quantization (Zhao et al., 2024). QAT can recover generation quality, but often requires substantial additional training resources (Zhang et al., 2026b). This raises a question: Can we design a single low-bit attention framework that supports training-free inference for models that tolerate quantization, and only lightweight fine-tuning for those that remain sensitive?

In this work, we present a **unified NVFP4 attention framework** that supports both training-free quantization and lightweight quantization-aware adaptation. Our key observation is that activation

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.

outliers and stable channel-wise biases amplify the distortion introduced by finite-precision quantization. We refer to this distribution-induced distortion as **structural quantization error**: outliers dominate shared block scales, while stable biases consume representable range that could otherwise resolve input-dependent variation. Both effects can be mitigated through inexpensive deterministic transformations. After these corrections, residual numerical error remains because FP4 values and low-precision block scales cannot represent the transformed activations exactly.

This distinction suggests a practical strategy for unifying PTQ and QAT: **reduce quantization distortion before learning to compensate for it**. We reshape attention activations through grouped rotation and calibrated centering, and match value and scale quantization to the resulting distributions. For models that tolerate the remaining error, this enables **training-free NVFP4 attention**. For more sensitive distilled models, lightweight QAT adapts to the residual distortion after these corrections, reducing the burden on fine-tuning. Both settings use the same hardware-efficient inference procedure.

We evaluate our method on Wan2.2 (Team Wan et al., 2025) and MiniMax H3 (MiniMax, 2026) across both conventional and distilled settings. On standard multi-step models, our training-free method preserves video quality close to 16-bit attention. Our fused kernels achieve a 2.11–$2.33\times$ **end-to-end attention speedup over FlashAttention** across sequence lengths from 8K to 42K; their preprocessing throughput is 1.18–$1.41\times$ **that of SageAttention3**. On the quantization-sensitive four-step Wan2.2 model, fine-tuning with only **30 randomly sampled examples** is sufficient to recover generation quality. On eight-step MiniMax H3, high generation quality can be maintained even without fine-tuning. These results show that suppressing structural sources of quantization error enables **accurate, efficient, and broadly applicable 4-bit attention** for modern video diffusion models.

Our main contributions are summarized as follows:

- We introduce a **unified NVFP4 attention framework** that suppresses structural sources of quantization error before model adaptation. This enables training-free inference for quantization-tolerant models and reduces the error that lightweight QAT must compensate for in sensitive models.
- We develop **hardware-efficient error suppression** that jointly addresses activation outliers, stable channel biases, and limited value and scale precision. Fused preprocessing and native NVFP4 attention kernels translate these accuracy improvements into efficient inference.
- We demonstrate near-16-bit generation quality across conventional and distilled video diffusion models, with a 2.11–$2.33\times$ **end-to-end attention speedup over FlashAttention** and adaptation using as few as **30 randomly sampled examples**.

2 Related Work

Low-bit attention. SageAttention (Zhang et al., 2025b) accelerates attention through INT8 query–key multiplication and key smoothing. SageAttention2 (Zhang et al., 2025a) combines INT4 query–key and FP8 probability–value multiplication with outlier smoothing, while SageAttention3 (Zhang et al., 2026a) extends both multiplications to NVFP4 with Q/K smoothing and two-level probability scaling. We use offline-calibrated Q/K offsets and fixed probability amplification to simplify runtime processing.

Quantization-aware attention training. Attn-QAT (Zhang et al., 2026b) stabilizes FP4 attention training by matching probability quantization across forward and backward passes and computing an auxiliary high-precision output for softmax gradients. We first suppress structural errors, then adapt only LoRA parameters with QKV fake quantization. Keeping probabilities unquantized during adaptation enables standard FlashAttention backward; probability quantization is introduced at inference.

Rotation for low-bit quantization. Orthogonal rotations redistribute outliers and improve low-bit quantization (Chee et al., 2023; Tseng et al., 2024a;b). Their effectiveness has been shown for model-wide quantization (Ashkboos et al., 2024), training-free and diffusion settings (Lin et al.,

2024; Choi et al., 2025; Huang et al., 2025; Yang et al., 2025), and learned or data-dependent transforms (Liu et al., 2024; Hu et al., 2025; Xu et al., 2025a; Sun et al., 2024; Xu et al., 2025b; Sharify et al., 2026). For microscaling FP4, prior work applies block rotations during training (NVIDIA et al., 2026; Chen et al., 2026) and to optimizer states (Ding et al., 2026), with further gains from group-matched rotations (Egiazarian et al., 2026) and outlier-aware microscaling designs (Lin et al., 2026). We use wider Q/K rotation groups and fold value-side transforms into projections.

Scale selection for NVFP4. Four Over Six (Cook et al., 2026) adaptively maps block maxima to 4 or 6, while ScaleSearch (Gupta et al., 2026) searches neighboring FP8 scales to minimize reconstruction error. ScaleSearchAttention also incorporates Q/K transformations. After rotation and centering, we find that a fixed target magnitude of 4 for QKV captures most of the benefit of per-block search in our video attention experiments, avoiding runtime candidate evaluation. We retain a target of 6 for unrotated probabilities.

3 PRELIMINARIES

3.1 BLOCK-WISE FP4 QUANTIZATION

We use the E2M1 FP4 format, whose representable values are

$$\{0, \pm 0.5, \pm 1, \pm 1.5, \pm 2, \pm 3, \pm 4, \pm 6\}.$$

For each 16-element quantization block, the scale and quantized values are computed as

$$s = \left\lfloor \frac{\max |x_i|}{r} \right\rceil_{\text{E4M3}}, \qquad \widehat{x}_i = \left\lfloor \frac{x_i}{s} \right\rceil_{\text{E2M1}}, \tag{1}$$

where the maximum is taken over all elements in the current block. Here, $\lfloor \cdot \rceil_{\text{E4M3}}$ and $\lfloor \cdot \rceil_{\text{E2M1}}$ denote rounding to the nearest value representable in the corresponding format. We write the block-wise quantization operation as

$$\left(\widehat{\mathbf{X}}, \boldsymbol{s}\right) = \mathcal{Q}_r(\mathbf{X}), \tag{2}$$

where $\boldsymbol{s}$ collects the scales of all blocks and $\widehat{\mathbf{X}}$ contains the quantized values. By default, we set $r = 6$, the largest finite magnitude representable in the E2M1 format.

3.2 ONLINE-SOFTMAX ATTENTION

Let $\mathbf{Q} \in \mathbb{R}^{B \times H \times N_q \times D}$ and $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{B \times H \times N_k \times D}$, where B, H, N_q, N_k, and D denote the batch size, number of heads, query length, key/value length, and head dimension, respectively. For query block $\mathbf{Q}_i$ and key/value blocks $\mathbf{K}_j$ and $\mathbf{V}_j$, the attention logits are $S_{ij} = \mathbf{Q}_i \mathbf{K}_j^{\mathsf{T}} / \sqrt{D}$. For each query row, the row-wise maximum is $m_i = \max_j S_{ij}$, the unnormalized attention weights are $P_{ij} = \exp(S_{ij} - m_i)$, and the normalization factor is $\ell_i = \sum_j P_{ij}$, where the maximum and sum are taken over all key positions. The output accumulator and normalized attention output are $\mathbf{O}_i = \sum_j P_{ij} \mathbf{V}_j$ and $\mathbf{O}_{\text{out},i} = \mathbf{O}_i / \ell_i$, respectively. FlashAttention performs these computations online by processing the key/value blocks sequentially and updating the row-wise maximum, normalization factor, and output accumulator without materializing the full attention matrix.

4 METHOD

Our FP4 attention framework first suppresses structural sources of quantization error to improve training-free inference, then uses lightweight QAT for models that remain sensitive. For a tensor $\mathbf{X}$, the quantization error is

$$\boldsymbol{\varepsilon}_X = \mathcal{D}\mathcal{Q}(\mathcal{Q}_r(\mathbf{X})) - \mathbf{X}, \tag{3}$$

where $\mathcal{Q}_r$ is the block-wise quantizer and $\mathcal{D}\mathcal{Q}$ dequantizes its output. We distinguish *structural errors* associated with activation outliers and stable channel-wise biases from *residual numerical errors* due to FP4 discretization and low-precision block scales.

Grouped Hadamard rotation and calibration-based channel centering reduce structural errors. QKV scale selection and probability amplification further reduce rounding error and scale underflow, respectively. Section 4.1 describes the inference procedure, and Section 4.2 presents QAT for adapting models to the remaining error.

4.1 FP4 Attention Inference

Our inference procedure combines grouped Hadamard rotation, quantization scale selection, calibration-based channel centering, and probability amplification, as summarized in Algorithm 1.

4.1.1 Grouped Hadamard Rotation

Rotation spreads outliers across channels and reduces the block maxima used to compute quantization scales. We apply the same orthogonal rotation to queries and keys, preserving the attention logits before quantization:

$$(\mathbf{QR})(\mathbf{KR})^\mathsf{T} = \mathbf{QRR}^\mathsf{T}\mathbf{K}^\mathsf{T} = \mathbf{QK}^\mathsf{T}, \qquad \mathbf{RR}^\mathsf{T} = \mathbf{I}. \tag{4}$$

We construct $\mathbf{R}$ from normalized Walsh–Hadamard matrices:

$$\mathbf{H}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad \mathbf{H}_{2^m} = \mathbf{H}_2^{\otimes m}, \tag{5}$$

where $\otimes$ denotes the Kronecker product. For a power-of-two group size g that divides the head dimension D, the grouped rotation is

$$\mathbf{R}_g = \mathbf{I}_{D/g} \otimes \mathbf{H}_g = \operatorname{diag}\left(\underbrace{\mathbf{H}_g, \ldots, \mathbf{H}_g}_{D/g \text{ groups}}\right). \tag{6}$$

Each outlier is spread over the g channels in its group, and the groups are rotated independently.

Why wide group rotation? With $g = 16$, rotation is confined to a single NVFP4 quantization block. It leaves neighboring blocks unchanged, but the channels in those blocks cannot help redistribute an outlier, so the maximum in the affected block can remain large. A wider group spreads outlier energy across multiple quantization blocks. This may increase the maxima of blocks with fewer outliers, but it reduces outlier peaks more effectively than rotation within a single quantization block.

In our experiments, a full rotation $\mathbf{H}_D$ and a two-group rotation achieve similar Q/K quantization accuracy and perform best among the tested configurations, outperforming 16-channel grouped rotation $\operatorname{diag}(\mathbf{H}_{16}, \ldots, \mathbf{H}_{16})$. We therefore choose

$$\mathbf{R}_q = \mathbf{R}_k = \operatorname{diag}\big(\mathbf{H}_{D/2}, \mathbf{H}_{D/2}\big), \tag{7}$$

which retains the benefit of spreading outliers across quantization blocks at a lower runtime cost than the full rotation. For $D = 128$, this gives two independent $\mathbf{H}_{64}$ transforms, each spanning four 16-channel quantization blocks.

For values, we use the full rotation $\mathbf{R}_V = \mathbf{H}_D$. The rotation and its inverse are absorbed into the value and output projections, respectively, so they require no additional computation during inference.

4.1.2 Quantization Scale Selection

For each 16-element block $\mathcal{B}$, we compute the E4M3 scale as $s_{X,\mathcal{B}} = \operatorname{cast}_{\mathrm{E4M3}}(\max_{i\in\mathcal{B}} |X_i|/r)$. After rotation, QKV magnitudes become more balanced. We set $r = 4$ to map the block maximum to 4 before scale rounding. This uses the more evenly spaced FP4 levels within $[0, 4]$ and avoids the large gap between the levels 4 and 6. In our experiments, per-block scale search yields little further improvement. For $\mathbf{P}$, which remains highly skewed without rotation, we retain $r = 6$ to use the full E2M1 range and preserve small probabilities.

4.1.3 Calibration-Based Channel Centering

Attention activations exhibit biases that vary by layer and channel. For $\mathbf{Q}$ and $\mathbf{K}$, these biases remain stable across prompts. We estimate a fixed offset for each channel by averaging over calibration samples and token positions, separately for each layer, timestep, and head.

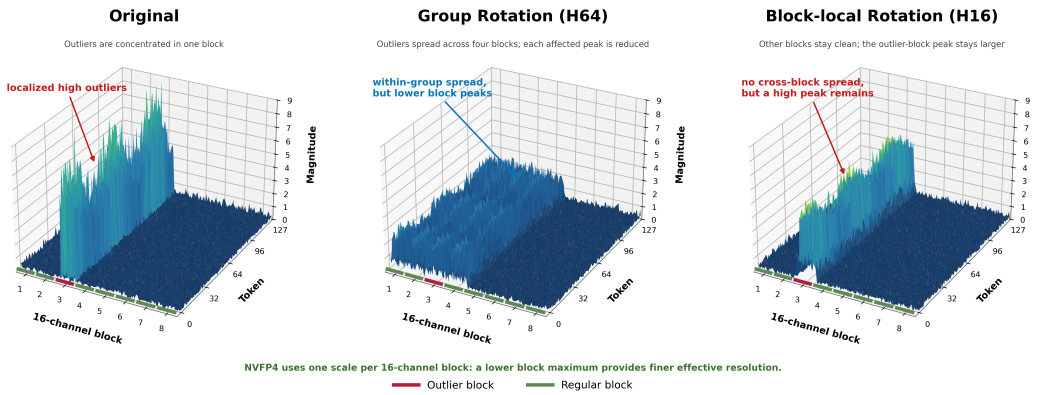

Figure 1: Hadamard rotation within and across 16-element NVFP4 quantization blocks. Rotation within a block keeps outliers in that block, where the maximum can remain large. Rotation across blocks reduces peak magnitudes by spreading outliers over more channels, including channels in blocks with smaller values.

Centering can be applied after rotation using precomputed rotated offsets:

$$(\mathbf{X} - \boldsymbol{\mu}_X)\mathbf{R}_X = \mathbf{X}\mathbf{R}_X - \underbrace{\boldsymbol{\mu}_X\mathbf{R}_X}_{\overline{\boldsymbol{\mu}}_X}. \tag{8}$$

This allows offset subtraction to be fused with rotation in the same matrix multiply-accumulate (MMA) operation, without additional computation. We write the centered tensor as $\widetilde{\mathbf{X}} = \mathbf{X}_r - \overline{\boldsymbol{\mu}}_X$, where $\mathbf{X}_r = \mathbf{X}\mathbf{R}_X$ and $\mathbf{X} \in \{\mathbf{Q}, \mathbf{K}\}$. Offsets are broadcast over token positions. The query–key product before centering can be written as

$$\mathbf{Q}_r\mathbf{K}_r^\mathsf{T} = \widetilde{\mathbf{Q}}\widetilde{\mathbf{K}}^\mathsf{T} + \underbrace{\overline{\boldsymbol{\mu}}_Q\widetilde{\mathbf{K}}^\mathsf{T}}_{\Delta\mathbf{S}} + \mathbf{Q}_r\overline{\boldsymbol{\mu}}_K^\mathsf{T}. \tag{9}$$

The last term is constant across keys for each query and does not affect softmax, since $\mathrm{softmax}(\mathbf{s} + c\mathbf{1}) = \mathrm{softmax}(\mathbf{s})$. We therefore need only the correction $\Delta\mathbf{S}$, broadcast across query positions. The corrected logits after quantization are

$$\widehat{\mathbf{S}} = \frac{\widehat{\widetilde{\mathbf{Q}}}\widehat{\widetilde{\mathbf{K}}}^\mathsf{T} + \Delta\mathbf{S}}{\sqrt{D}}. \tag{10}$$

We compute $\Delta\mathbf{S}$ from the centered keys before FP4 quantization, using higher precision. Key rotation, offset subtraction, correction computation, and quantization are fused into one preprocessing kernel. In some models, the channel biases in $\mathbf{V}$ vary substantially across prompts, making fixed offsets unreliable. We therefore do not apply fixed offsets to $\mathbf{V}$ and use only the Hadamard rotation described above before quantization.

4.1.4 Probability Amplification

Let $\mathbf{P} = \exp(\mathbf{S} - m')$ denote the unnormalized attention weights. When $\mathbf{P}$ is quantized directly to NVFP4, its block scales occupy only a small range near zero in E4M3. We multiply $\mathbf{P}$ by $\alpha = 6 \times 128 = 768$, giving $\overline{\mathbf{P}} = \alpha\mathbf{P}$. Before conversion to E4M3, the block scales change from

$$a_{P,\mathcal{B}} = \frac{\max_{i\in\mathcal{B}} P_i}{6} \in [0, 1/6] \quad \text{to} \quad a_{\overline{P},\mathcal{B}} = \frac{\max_{i\in\mathcal{B}} \overline{P}_i}{6} = 128 \max_{i\in\mathcal{B}} P_i \in [0, 128]. \tag{11}$$

This uses a wider range of E4M3 values and reduces the risk of small block scales rounding to zero. Before quantization, amplification leaves the normalized attention output unchanged: it scales the output accumulator $\mathbf{O}$ and normalization factor $\boldsymbol{\ell}$ by the same factor, which cancels in the final division:

$$\overline{\mathbf{O}} = \alpha\mathbf{O}, \qquad \overline{\boldsymbol{\ell}} = \alpha\boldsymbol{\ell}, \qquad \mathbf{O}_{\text{out}} = \frac{\overline{\mathbf{O}}}{\overline{\boldsymbol{\ell}}} = \frac{\mathbf{O}}{\boldsymbol{\ell}}. \tag{12}$$

Algorithm 1 FP4 Attention Inference

Offline setup: Fix $\mathbf{R}_q$, $\mathbf{R}_k$, and $\mathbf{R}_V$; calibrate $\overline{\boldsymbol{\mu}}_Q$ and $\overline{\boldsymbol{\mu}}_K$
Absorb $\mathbf{W}'_V = \mathbf{W}_V\mathbf{R}_V$ and $\mathbf{W}'_O = \mathbf{R}_V^\mathsf{T}\mathbf{W}_O$
Input: $\mathbf{Q}, \mathbf{K}, \mathbf{V}$
Preprocessing:
$\widetilde{\mathbf{Q}} \leftarrow \mathbf{Q}\mathbf{R}_q - \mathbf{1}_{N_q}\overline{\boldsymbol{\mu}}_Q$; $(\widehat{\mathbf{Q}}, \mathbf{s}_Q) \leftarrow \mathcal{Q}_4(\widetilde{\mathbf{Q}})$
$\widetilde{\mathbf{K}} \leftarrow \mathbf{K}\mathbf{R}_k - \mathbf{1}_{N_k}\overline{\boldsymbol{\mu}}_K$; $\Delta\mathbf{S} \leftarrow \overline{\boldsymbol{\mu}}_Q\widetilde{\mathbf{K}}^\mathsf{T}$; $(\widehat{\mathbf{K}}, \mathbf{s}_K) \leftarrow \mathcal{Q}_4(\widetilde{\mathbf{K}})$
$(\widehat{\mathbf{V}}, \mathbf{s}_V) \leftarrow \mathcal{Q}_4(\mathbf{V})$ ▷ $\mathbf{V}$ rotated by $\mathbf{W}'_V$
NVFP4 attention kernel:
Load blockwise: $\{(\widehat{\mathbf{Q}}_i, \mathbf{s}_{Q,i})\}_{i=1}^{T_q}$ and $\{(\widehat{\mathbf{K}}_j, \mathbf{s}_{K,j}, \widehat{\mathbf{V}}_j, \mathbf{s}_{V,j}, \Delta\mathbf{S}_j)\}_{j=1}^{T_{kv}}$
for $i = 1$ to T_q **do**
  $m_i \leftarrow -\infty$; $\overline{\boldsymbol{\ell}}_i \leftarrow \mathbf{0}$; $\overline{\mathbf{O}}_i \leftarrow \mathbf{0}$
  for $j = 1$ to T_{kv} **do**
    $\mathbf{S}_{i,j} \leftarrow \frac{1}{\sqrt{D}}\,\mathrm{NVFP4MM}\Big(\widehat{\mathbf{Q}}_i, \mathbf{s}_{Q,i}, \widehat{\mathbf{K}}_j, \mathbf{s}_{K,j}, \Delta\mathbf{S}_j\Big)$
    $m'_i \leftarrow \max(m_i, \mathrm{rowmax}(\mathbf{S}_{i,j}))$; $\beta_i \leftarrow \exp(m_i - m'_i)$
    $\overline{\mathbf{P}}_{i,j} \leftarrow \exp(\mathbf{S}_{i,j} - (m'_i - \log\alpha))$; $(\widehat{\overline{\mathbf{P}}}_{i,j}, \mathbf{s}_{\overline{P},i,j}) \leftarrow \mathcal{Q}_6(\overline{\mathbf{P}}_{i,j})$
    $\overline{\boldsymbol{\ell}}_i \leftarrow \beta_i\overline{\boldsymbol{\ell}}_i$; $\overline{\mathbf{O}}_i \leftarrow \beta_i\overline{\mathbf{O}}_i$; $m_i \leftarrow m'_i$ ▷ rescale and update m
    $\overline{\boldsymbol{\ell}}_i \mathrel{+}= \mathrm{rowsum}(\overline{\mathbf{P}}_{i,j})$; $\overline{\mathbf{O}}_i \mathrel{+}= \mathrm{NVFP4MM}\Big(\widehat{\overline{\mathbf{P}}}_{i,j}, \mathbf{s}_{\overline{P},i,j}, \widehat{\mathbf{V}}_j, \mathbf{s}_{V,j}\Big)$
  end for
  $\mathbf{O}_{\mathrm{out},i} \leftarrow \overline{\mathbf{O}}_i/\overline{\boldsymbol{\ell}}_i$ ▷ $\mathbf{R}_V^\mathsf{T}$ absorbed into $\mathbf{W}'_O$
end for
return $\mathbf{O}_{\mathrm{out}}$

We implement amplification by shifting the softmax exponent: $\overline{\mathbf{P}} = \exp(\mathbf{S} - (m' - \log\alpha))$. This requires no additional multiplication and adds negligible runtime overhead.

Models sensitive to the remaining quantization error are adapted with QAT (Section 4.2) and then deployed with the same inference procedure.

4.2 Quantization-Aware Training

Aggressively distilled few-step models can remain sensitive to residual FP4 error after rotation and centering. We address this with LoRA-based QAT (Hu et al., 2021): training simulates the inference-time QKV quantizers and updates only the LoRA parameters, while the base model, rotations, calibrated offsets, and other quantization settings remain fixed.

4.2.1 Learning the Residual Numerical Error

Grouped Hadamard rotation and calibration-based channel centering reduce outlier- and bias-induced errors before quantization. With these structural corrections fixed, QAT focuses on learning to compensate for the residual numerical error. This narrows the learning task and reduces the burden on model adaptation.

We simulate FP4 quantization by quantizing QKV to FP4 and dequantizing them to the training precision, either FP16 or BF16:

$$\mathcal{Q}_4(\mathbf{X}) = (\widehat{\mathbf{X}}, \mathbf{s}), \qquad \mathcal{DQ}(\widehat{\mathbf{X}}, \mathbf{s}) = \mathbf{s}_\mathcal{B} \odot \widehat{\mathbf{X}}, \tag{13}$$

where $\mathbf{s}_\mathcal{B}$ broadcasts each block scale to the shape of $\widehat{\mathbf{X}}$. The resulting fake quantization operation is $\mathcal{FQ}_4(\mathbf{X}) \equiv \mathcal{DQ}(\mathcal{Q}_4(\mathbf{X}))$. To pass gradients through rounding, we use a straight-through estimator (STE), treating fake quantization as the identity during backpropagation: $\partial\mathcal{FQ}_4(\mathbf{X})/\partial\mathbf{X} = \mathbf{1}$.

For Q/K, we fake-quantize the rotated and centered tensors, then add back their fixed offsets in FP16 or BF16:

$$\widehat{\mathbf{X}}_c = \mathcal{FQ}_4(\mathbf{X}\mathbf{R}_X - \mathbf{1}_{N_X}\overline{\boldsymbol{\mu}}_X), \qquad \widehat{\mathbf{X}}_{\mathrm{QAT}} = \widehat{\mathbf{X}}_c + \mathbf{1}_{N_X}\overline{\boldsymbol{\mu}}_X, \quad \mathbf{X} \in \{\mathbf{Q}, \mathbf{K}\}. \tag{14}$$

Algorithm 2 QAT Forward Pass

Offline setup: Calibrate $\overline{\boldsymbol{\mu}}_Q$ and $\overline{\boldsymbol{\mu}}_K$
Input: $\mathbf{Q}, \mathbf{K}, \mathbf{V}$

$\widetilde{\mathbf{Q}} \leftarrow \mathbf{Q}\mathbf{R}_q - \mathbf{1}_{N_q}\overline{\boldsymbol{\mu}}_Q$
$\widetilde{\mathbf{K}} \leftarrow \mathbf{K}\mathbf{R}_k - \mathbf{1}_{N_k}\overline{\boldsymbol{\mu}}_K$
$\widetilde{\mathbf{V}} \leftarrow \mathbf{V}\mathbf{R}_V$ ▷ online rotate
$\widehat{\mathbf{Q}}_c \leftarrow \mathcal{FQ}_4(\widetilde{\mathbf{Q}}), \quad \widehat{\mathbf{K}}_c \leftarrow \mathcal{FQ}_4(\widetilde{\mathbf{K}}), \quad \widehat{\mathbf{V}}_c \leftarrow \mathcal{FQ}_4(\widetilde{\mathbf{V}})$
$\widehat{\mathbf{Q}}_{\mathrm{QAT}} \leftarrow \widehat{\mathbf{Q}}_c + \mathbf{1}_{N_q}\overline{\boldsymbol{\mu}}_Q$
$\widehat{\mathbf{K}}_{\mathrm{QAT}} \leftarrow \widehat{\mathbf{K}}_c + \mathbf{1}_{N_k}\overline{\boldsymbol{\mu}}_K$
$\mathbf{O}_r \leftarrow \mathrm{FlashAttention}\Big(\widehat{\mathbf{Q}}_{\mathrm{QAT}}, \widehat{\mathbf{K}}_{\mathrm{QAT}}, \widehat{\mathbf{V}}_c\Big)$
$\mathbf{O}_{\mathrm{out}} \leftarrow \mathbf{O}_r\mathbf{R}_V^{\mathsf{T}}$ ▷ online rotate
return $\mathbf{O}_{\mathrm{out}}$

This removes the fixed biases before quantization while allowing the model to adapt to the resulting numerical error. Adding the offsets back makes standard attention include the correction $\Delta\mathbf{S}_{\mathrm{QAT}} = \overline{\boldsymbol{\mu}}_Q\widehat{\mathbf{K}}_c^{\mathsf{T}}$, broadcast across queries. Training computes this correction from fake-quantized centered keys, whereas inference uses centered keys before quantization. The two corrections therefore differ by a residual quantization term. This formulation supports standard FlashAttention during QAT without a custom backward implementation.

4.2.2 QAT Forward Pass

Our experiments show that QKV quantization accounts for most of the video-quality degradation, while quantizing the attention probabilities $\mathbf{P}$ has only a small effect. We therefore fake-quantize only QKV during training and keep $\mathbf{P}$ in FP16 or BF16. This focuses training on the main source of error and supports the standard FlashAttention backward pass. Probability quantization and amplification are used only during inference. During training, we apply the value rotation and its inverse explicitly; after training, we absorb them into the updated value and output projections. The Hadamard rotation is symmetric, $\mathbf{R}_V^{\mathsf{T}} = \mathbf{R}_V$. Algorithm 2 summarizes the forward pass.

5 Experiments

5.1 Experimental Setup

Implementation and baselines. We implement our kernels in CUDA. Inference speed and video quality are evaluated on NVIDIA RTX PRO 6000D GPUs, while QAT runs on NVIDIA H200 GPUs. Speed baselines are SageAttention3 (Zhang et al., 2026a), Attn-QAT (Zhang et al., 2026b), and FlashAttention. We report preprocessing and end-to-end throughput for non-causal MHA with $H = 24$, $D = 128$, and sequence lengths from 8K to 42K.

Models and generation settings. We evaluate the original and distilled variants of Wan2.2 (Team Wan et al., 2025) and MiniMax H3 (MiniMax, 2026). All Wan2.2 experiments use T2V generation at 720p; the original model uses 50 denoising steps, while `Wan2.2-Lightning` uses four (LightX2V, 2025). The original 40-step MiniMax H3 generates 144 frames at 544p, whereas the eight-step `MiniMax-H3-Acc-LoRAs` variant is evaluated at 720p (Alibaba PAI, 2026).

Mixed-precision fallback. We retain FP16 attention only in the most quantization-sensitive blocks: the first two and last two blocks of Wan2.2, and the final five blocks of MiniMax H3. All remaining attention blocks use RotateAttention-FP4.

Training-free evaluation. Each FP4 output is compared with its paired FP16 reference using cosine similarity, MSE, SSIM, PSNR, and the seven-dimension VBench (Huang et al., 2023) average. MiniMax H3 uses a fixed fox prompt for VBench, and the Wan2.2 VBench average is computed from two videos. No fine-tuning is used beyond offline offset calibration.

Quantization-aware adaptation. On four-step Wan2.2-T2V, we compare our LoRA-based QAT (Section 4.2) with training-free FP4 and Attn-QAT using FP16 as the reference. Both QAT methods use 30 videos at 720p and 120 updates, split equally between the two experts.

5.2 Results

5.2.1 Inference Efficiency

Figure 2 reports kernel throughput. RotateAttention-FP4 reaches 299–324 TOPS end to end, a 2.11–2.33× speedup over FlashAttention and a 0.3–1.9% gain over SageAttention3. Preprocessing reaches 2,581–13,706 TOPS, or 1.18–1.41× SageAttention3. Attn-QAT is faster at shorter sequences, but its end-to-end lead narrows from 10.5% at 8K to 0.7% at 42K; our preprocessing is 9.7% faster at 42K.

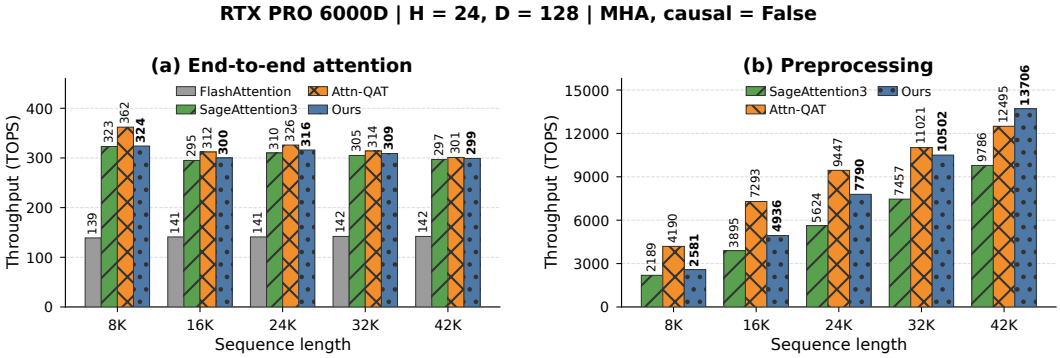

Figure 2: **Inference throughput on NVIDIA RTX PRO 6000D.** End-to-end attention and preprocessing for non-causal MHA with $H = 24$ and $D = 128$. Higher is better.

5.2.2 Training-Free Generation Quality

Table 1 compares training-free RotateAttention-FP4 with SageAttention3 using FP16 generations as the paired reference. Our method reduces MSE by 68.2%, 19.1%, and 28.0% on 40-step MiniMax H3, eight-step MiniMax H3, and Wan2.2-T2V, respectively, lowering it from 1094.50 to 347.70, 1154.63 to 934.21, and 1650.14 to 1187.59. Cosine similarity, SSIM, and PSNR also improve in every setting. In contrast, absolute VBench differences remain small (0.0001–0.0127), indicating better FP16 fidelity with comparable generation quality.

Table 1: **Training-free quality relative to FP16.** Paired-reference metrics compare each FP4 output with FP16; VBench Avg. averages seven dimensions. Bold marks the better FP4 result.

Model	Method	Cosine ↑	MSE ↓	SSIM ↑	PSNR (dB) ↑	VBench Avg. ↑
MiniMax H3 (40-step)	FP16 reference	–	–	–	–	0.7830
	Ours (FP4)	**0.993447**	**347.70**	**0.8204**	**23.04**	0.7818
	SageAttention3	0.979319	1094.50	0.6858	17.81	**0.7819**
MiniMax H3 (8-step)	FP16 reference	–	–	–	–	0.7758
	Ours (FP4)	**0.979434**	**934.21**	**0.730030**	**18.43**	0.7747
	SageAttention3	0.974925	1154.63	0.722126	17.52	**0.7788**
Wan2.2-T2V	FP16 reference	–	–	–	–	0.7712
	Ours (FP4)	**0.959349**	**1187.59**	**0.6929**	**18.26**	0.6880
	SageAttention3	0.941546	1650.14	0.6908	17.23	**0.7007**

5.2.3 Quantization-Aware Adaptation on Four-Step Wan2.2-T2V

Training dynamics. Figure 3 compares conditional NMSE during QAT. For the high-noise expert, our method exhibits fewer spikes and lowers the final loss from 0.0408 to 0.0184 (54.9%), indicating more stable adaptation. For the low-noise expert, the final losses remain comparable: 0.0169 for ours and 0.0163 for Attn-QAT.

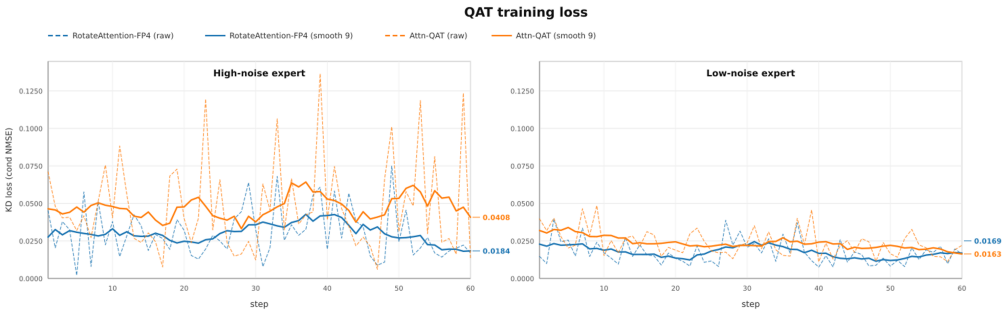

Figure 3: **QAT dynamics on four-step Wan2.2-T2V.** Conditional NMSE for the high-noise (left) and low-noise (right) experts. Dashed curves are raw losses; solid curves use nine-step smoothing. Each expert receives 60 updates.

Video quality after adaptation. Table 2 compares both methods under the same 120-step budget. Ours improves background consistency, aesthetic quality, imaging quality, and motion smoothness, while tying dynamic degree. The largest gain is in aesthetic quality (0.7009 to 0.7273), and the overall score increases from 0.8607 to 0.8669.

Table 2: **VBench on four-step Wan2.2-T2V.** Both methods use 30 samples and 120 updates. Overall averages six dimensions; higher is better and bold marks the best result.

Model	Method	Subject cons.	Background cons.	Aesthetic quality	Imaging quality	Motion smooth.	Dynamic degree	Overall
Wan2.2-T2V (4-step)	Ours	0.9569	**0.9692**	**0.7273**	**0.7055**	**0.9855**	**0.8571**	**0.8669**
	Attn-QAT	**0.9577**	0.9610	0.7009	0.7022	0.9851	**0.8571**	0.8607

Overall comparison. The training-free results show that structural error suppression is sufficient for both original and distilled models under the stated mixed-precision policies: paired-reference errors decrease consistently while VBench remains close to the baseline. For quantization-sensitive four-step Wan2.2, the same inference formulation supports lightweight adaptation; under an equal data and update budget, it stabilizes high-noise training and improves overall VBench relative to Attn-QAT. These results support training-free inference when residual error is tolerable and low-cost QAT otherwise, without changing the deployment path.

6 Conclusion

We presented RotateAttention-FP4, a unified NVFP4 attention framework that suppresses structural quantization error before adaptation. Grouped rotation, calibrated channel centering, scale selection, and probability amplification enable accurate training-free inference, while lightweight LoRA QAT handles the remaining error in quantization-sensitive models using the same inference procedure. Across Wan2.2 and MiniMax H3, our training-free method improves fidelity to FP16 over SageAttention3 while retaining comparable VBench quality. With only 30 training samples and 120 updates, the QAT variant also surpasses Attn-QAT on four-step Wan2.2. Together with a 2.11–2.33$\times$ end-to-end speedup over FlashAttention, these results demonstrate an accurate and efficient path to 4-bit attention for video generation. Further reducing the remaining FP16 fallback in sensitive boundary blocks is a promising direction toward fully quantized deployment across broader prompts and resolutions.

REFERENCES

Alibaba PAI. MiniMax-H3-Acc-LoRAs. Hugging Face model card and checkpoints, 2026. URL https://huggingface.co/alibaba-pai/MiniMax-H3-Acc-LoRAs. Eight-step acceleration LoRAs for MiniMax H3. Accessed September 26, 2026.

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, 2024. URL https://arxiv.org/abs/2404.00456.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. QuIP: 2-Bit Quantization of Large Language Models With Guarantees, 2023. URL https://arxiv.org/abs/2307.13304.

Yuxiang Chen, Yifan Liu, Xiaoming Xu, Pengle Zhang, Michael Beyer, Martin Rapp, Jun Zhu, and Jianfei Chen. TetraJet-v2: Accurate NVFP4 Training for Large Language Models with Oscillation Suppression and Outlier Control, 2026. URL https://arxiv.org/abs/2510.27527.

Euntae Choi, Sumin Song, Woosang Lim, and Sungjoo Yoo. Grouped Sequency-arranged Rotation: Optimizing Rotation Transformation for Quantization for Free, 2025. URL https://arxiv.org/abs/2505.03810.

Jack Cook, Junxian Guo, Guangxuan Xiao, Yujun Lin, Keith Wyss, Mahdi Nazemi, Asit Mishra, Carlo del Mundo, Tijmen Blankevoort, and Song Han. Four Over Six: More Accurate NVFP4 Quantization with Adaptive Block Scaling, 2026. URL https://arxiv.org/abs/2512.02010.

Siyu Ding, Mingchuan Ma, Jiabo Tong, Xingrun Xing, Ziming Wang, and Guoqi Li. Full-Stack FP4: Stable LLM Pretraining with Quantized Projections, Optimizers, and Attention, 2026. URL https://arxiv.org/abs/2607.04422.

Vage Egiazarian, Roberto L. Castro, Denis Kuznedelev, Andrei Panferov, Eldar Kurtic, Shubhra Pandit, Alexandre Marques, Mark Kurtz, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Bridging the Gap Between Promise and Performance for Microscaling FP4 Quantization, 2026. URL https://arxiv.org/abs/2509.23202.

Tanmaey Gupta, Hayden Prairie, Xiaoxia Wu, Reyna Abhyankar, Qingyang Wu, Austin Silveria, Pragaash Ponnusamy, Jue Wang, Ben Athiwaratkun, Leon Song, Tri Dao, Daniel Y. Fu, and Chris De Sa. Search Your Block Floating Point Scales!, 2026. URL https://arxiv.org/abs/2605.12464.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models, 2021. URL https://arxiv.org/abs/2106.09685.

Xing Hu, Yuan Cheng, Dawei Yang, Zukang Xu, Zhihang Yuan, Jiangyong Yu, Chen Xu, Zhe Jiang, and Sifan Zhou. OSTQuant: Refining Large Language Model Quantization with Orthogonal and Scaling Transformations for Better Distribution Fitting, 2025. URL https://arxiv.org/abs/2501.13987.

Feice Huang, Zuliang Han, Xing Zhou, Yihuang Chen, Lifei Zhu, and Haoqian Wang. ConvRot: Rotation-Based Plug-and-Play 4-bit Quantization for Diffusion Transformers, 2025. URL https://arxiv.org/abs/2512.03673.

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

LightX2V. Wan2.2-Lightning. Hugging Face model card and checkpoints, 2025. URL https://huggingface.co/lightx2v/Wan2.2-Lightning. Four-step distilled models. Accessed September 26, 2026.

Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. DuQuant: Distributing Outliers via Dual Transformation Makes Stronger Quantized LLMs, 2024. URL https://arxiv.org/abs/2406.01721.

Haokun Lin, Xinle Jia, Haobo Xu, Bingchen Yao, Xianglong Guo, Yichen Wu, Zhichao Lu, Ying Wei, Qingfu Zhang, and Zhenan Sun. DuQuant++: Fine-grained Rotation Enhances Microscaling FP4 Quantization, 2026. URL https://arxiv.org/abs/2604.17789.

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, 2024. URL https://arxiv.org/abs/2405.16406.

MiniMax. MiniMax H3. Official code and model release, 2026. URL https://github.com/MiniMax-AI/MiniMax-H3. Accessed September 26, 2026.

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

Sayeh Sharify, Mahsa Salmani, and Hesham Mostafa. DiRotQ: Rotation-Aware Quantization for 4-bit Diffusion Transformers, 2026. URL https://arxiv.org/abs/2605.16732.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, Xin Jiang, Wulong Liu, and Jun Yao. FlatQuant: Flatness Matters for LLM Quantization, 2024. URL https://arxiv.org/abs/2410.09426.

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. URL https://arxiv.org/abs/2503.20314.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP#: Even Better LLM Quantization with Hadamard Incoherence and Lattice Codebooks, 2024a. URL https://arxiv.org/abs/2402.04396.

Albert Tseng, Qingyao Sun, David Hou, and Christopher De Sa. QTIP: Quantization with Trellises and Incoherence Processing, 2024b. URL https://arxiv.org/abs/2406.11235.

Bingxin Xu, Zhen Dong, Oussama Elachqar, and Yuzhang Shang. ButterflyQuant: Ultra-low-bit LLM Quantization through Learnable Orthogonal Butterfly Transforms, 2025a. URL https://arxiv.org/abs/2509.09679.

Zukang Xu, Yuxuan Yue, Xing Hu, Zhihang Yuan, Zixu Jiang, Zhixuan Chen, Jiangyong Yu, Chen Xu, Sifan Zhou, and Dawei Yang. MambaQuant: Quantizing the Mamba Family with Variance Aligned Rotation Methods, 2025b. URL https://arxiv.org/abs/2501.13484.

Lianwei Yang, Haokun Lin, Tianchen Zhao, Yichen Wu, Hongyu Zhu, Ruiqi Xie, Zhenan Sun, Yu Wang, and Qingyi Gu. LRQ-DiT: Log-Rotation Post-Training Quantization of Diffusion Transformers for Image and Video Generation, 2025. URL https://arxiv.org/abs/2508.03485.

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, 2025a. URL https://arxiv.org/abs/2411.10958.

Jintao Zhang, Jia Wei, Haofeng Huang, Pengle Zhang, Jun Zhu, and Jianfei Chen. SageAttention: Accurate 8-Bit Attention for Plug-and-play Inference Acceleration, 2025b. URL https://arxiv.org/abs/2410.02367.

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, 2026a. URL https://arxiv.org/abs/2505.11594.

Peiyuan Zhang, Matthew Noto, Wenxuan Tan, Chengquan Jiang, Will Lin, Wei Zhou, and Hao Zhang. Attn-QAT: 4-Bit Attention With Quantization-Aware Training, 2026b. URL https://arxiv.org/abs/2603.00040.

Tianchen Zhao, Xuefei Ning, Tongcheng Fang, Enshu Liu, Guyue Huang, Zinan Lin, Shengen Yan, Guohao Dai, and Yu Wang. MixDQ: Memory-Efficient Few-Step Text-to-Image Diffusion Models with Metric-Decoupled Mixed Precision Quantization, 2024. URL https://arxiv.org/abs/2405.17873.