DATA-ADAPTIVE TRANSFORMS CAN IMPROVE THE ACCURACY OF QUANTIZATION-AWARE TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Performing large model training in lower precision, a process known as quantization-aware training (QAT), has become a standard approach for reducing costs, both for inference (since it produces a smaller model) and training time (since the matrix multiplications can be performed faster). However, training models accurately in lower precision is still a challenge. In this work, we propose a new, more accurate approach for QAT, based on the idea of adaptively learning a set of "smoothing" matrices for each layer's matrix multiplication, whose goal is to specifically reduce its quantization error. We show for the first time that this adaptation can be performed efficiently and stably during training, leading to significant reductions in the accuracy gap for QAT for pre-training, fine-tuning, and distillation of quantized large language models.

1 INTRODUCTION

Quantization-aware training (QAT) trains large models with low-precision weights and activations (Jacob et al., 2018; Liu et al., 2023), becoming a standard way to reduce costs. It produces low-bitwidth models that are cheaper to serve (Xin et al., 2026; Counathe et al., 2026). Modern hardware such as NVIDIA Blackwell GPUs (NVIDIA, 2024) runs 4-bit matrix multiplications at two to three times the throughput of FP8 (NVIDIA, 2025), so QAT can speed up both inference and training itself.

Currently, the "bitwidth barrier" to accurate training lies at around 4 bits for weights and activations. Elaborate state of the art recipes (almost) recover accuracy in 4-bit pre-training and fine-tuning (Panferov et al., 2025; Castro et al., 2025; Panferov et al., 2026; Chen et al., 2025b; Cook et al., 2025; NVIDIA, 2025; Ashkboos et al., 2025). These recipes usually employ two ingredients: 1) use of *microscaling* formats, in which groups of 16–32 values share a scale (Rouhani et al., 2023); and 2) *fixed* or randomized Hadamard rotations, which smoothen outlier values before rounding. Yet, going below the 4-bit frontier or using coarser scales, training can become unstable. By contrast, the state of the art in post-training quantization (PTQ), is given by *data-adaptive* transforms fitted to each layer, which beat fixed rotations (Xiao et al., 2023; Liu et al., 2025; Sun et al., 2024b; Chen et al., 2026). As of yet, it is unclear whether this advantage carries over to *training*, as adaptive transforms raise significant stability and runtime cost issues.

Contribution. In this paper, we propose TWIST, a QAT method that adaptively learns, during training, a pair of "smoothing" matrices for each quantized linear layer's matrix multiplication, reducing quantization error. TWIST can apply these matrices in both the forward and the backward passes. We show, for the first time, that this adaptation can be performed *efficiently* and *stably* during training, reducing the accuracy gap of QAT in both pre-training and fine-tuning.

TWIST works by splitting the input dimension of each linear layer into blocks, and computes a dynamic, at-runtime, equivalent of the post-training WUSH transform of Chen et al. (2026), for each matrix block. In short, WUSH is a data-adaptive transform that can be pre-multiplied into both weights and activations *before quantization*, in order to minimize the quantization error of the matrix multiplication. Bringing this transform into training requires several new techniques: *(i) Online updates.* We show that the weight and activation statistics that determine the optimal transform drift slowly enough that TWIST only needs to recompute the transform every $\tau = 20$ steps, based on previous-step activation statistics. *(ii) Transform freezes.* However, the WUSH transform is determined only up to a rotation, so each recomputation nevertheless changes the quantization basis substantially. Towards the end of training, this causes a *basis shift* phenomenon: each refresh moves the quantization grid while the model is trying to settle into a local minimum. To address this, we

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.

freeze the transform later in training, and keep it fixed for fine-tuning. *(iii) Backward pass.* We show for the first time that adaptive transforms can be applied stably during back-propagation, and provide a way to do so while preserving unbiased gradients. *(iv) Low cost.* Since the transform is block-diagonal, applying it costs the same as a block rotation at runtime, and building it requires one Cholesky factorization and one SVD per block every τ steps. We provide an efficient implementation of these two steps.

Theoretical guarantees. We provide theoretical grounding for our method via a QAT analysis with full-precision master weights and straight-through gradients (Bengio et al., 2013), as a perturbed-iterate method. For linear regression we show that fixed-transform SGD–STE converges at rate $O(1/K)$ to a step-size-independent excess loss (Theorem E.8). This excess is bounded by the reconstruction error at the regression target, which WUSH minimizes, with equality under weight-only quantization. Second, for smooth nonlinear models with weight-only quantization, we show a risk bound that only depends on a single quantization term with a step-size-independent coefficient (Theorem E.10).

Results. We evaluate TWIST in three settings: pre-training from scratch, where we quantize the forward pass and, separately, the backward pass; quantization-aware distillation (QAD); and QAT fine-tuning. We compare it against QAT with no transform (Identity) and with a Hadamard transform, used by prior methods, e.g. (Panferov et al., 2025). We summarize our findings as follows:

- *Pre-training, forward pass.* We train Llama-style models from 200M to 3B parameters on C4, with weights and activations quantized to 4 bits (W4A4) or 3 bits (W3A3). Relative to Hadamard, TWIST closes 15–44% of the gap to full precision at every model size and both precisions. At 3B, it closes 37% of the gap at W4A4 and 44% at W3A3.
- *Pre-training, backward pass.* When the gradient with respect to the activations is quantized to 3 bits, TWIST with a randomized Hadamard factor reduces the resulting loss degradation by 12.5–20% relative to the standard randomized Hadamard transform, at 430M and 800M parameters.
- *Quantization-aware distillation (QAD) and QAT fine-tuning.* We apply QAD healing to Qwen3-4B-Thinking and Qwen3-32B (Yang et al., 2025) in INT4 and INT3, with weight-only and weight–activation quantization. TWIST reaches the lowest KL divergence to the teacher in all configurations, 8–27% below Hadamard. We also fine-tune Qwen3-4B-Base and Qwen3-14B-Base on mathematical reasoning (Moshkov et al., 2025). At INT3 with weight–activation quantization, TWIST reaches 55.8% (4B) and 69.6% (14B) on MATH-500. Hadamard QAT reaches 53.8% and 67.4%, and BF16 fine-tuning followed by PTQ only 15.2% and 51.6%.

Overall, our results show for the first time that data-adaptive transforms can be adapted efficiently and stably during training, substantially narrowing the accuracy gap of QAT at 3 and 4 bits.

2 Related work

Transforms for PTQ. Low-bit quantization of LLMs is difficult due to outlier features (Dettmers et al., 2022; Sun et al., 2024a), and PTQ methods often address them via invertible transforms, applied to both operands of each matrix multiplication. SmoothQuant (Xiao et al., 2023) uses per-channel scaling to migrate activation outliers, and AWQ (Lin et al., 2024b) uses them to protect salient weight channels. OmniQuant (Shao et al., 2024) and AffineQuant (Ma et al., 2024b) learn these scalings, while QuIP (Chee et al., 2023) introduced incoherence processing with orthogonal matrices. QuIP# (Tseng et al., 2024) made it efficient with randomized Hadamard transforms, and QuaRot (Ashkboos et al., 2024) extended it to weight–activation quantization by folding Hadamard rotations, while SpinQuant (Liu et al., 2025) learns the foldable rotations. Methods like DuQuant (Lin et al., 2024a), OSTQuant (Hu et al., 2025), and FlatQuant (Sun et al., 2024b) learn different types of transforms for PTQ. WUSH (Chen et al., 2026) derived the *optimal* block transform in closed form, for a bound on the layer's quantization error. All of these methods fit the transform to a *fixed* model.

Transforms during training. QAT methods have so far used data-independent transforms. Xi et al. (2023) proposed a block-diagonal Hadamard quantizer for 4-bit training, while QuEST (Panferov et al., 2025), HALO (Ashkboos et al., 2025), Quartet (Castro et al., 2025), and BitNet v2 (Wang et al., 2025) used fixed Hadamard rotations. Randomized Hadamard transforms keep the gradient estimates of low-precision backward passes unbiased (Tseng et al., 2025; Panferov et al., 2026; NVIDIA, 2025). The closest work to ours is RoSTE (Wei et al., 2025), which applies rotations during quantization-aware fine-tuning. However, RoSTE selects each rotation only among random

Walsh–Hadamard matrices; the rotations are not fitted to the layer statistics. GaugeQuant (Bento & Seabra, 2026) learns rotations during full-precision training, using the learned basis for PTQ.

Quantization-aware training. QAT simulates quantization in the forward pass and propagates gradients through the rounding with the straight-through estimator (STE) (Bengio et al., 2013; Courbariaux et al., 2015). The first QAT methods pushed this recipe to binary and low-bit networks (Hubara et al., 2016; Rastegari et al., 2016; Zhou et al., 2016), and to integer-only inference (Jacob et al., 2018). Subsequent methods learned the clipping ranges and step sizes (Choi et al., 2018; Esser et al., 2020). Nagel et al. (2022) showed that STE-trained weights oscillate between neighbouring grid points, a related observation to the basis shift. For LLMs, LLM-QAT (Liu et al., 2023) and EfficientQAT (Chen et al., 2024) adapted QAT to pre-trained models. BitNet (Wang et al., 2023) and BitNet b1.58 (Ma et al., 2024a) trained binary and ternary-weight LLMs from scratch via complex heuristics. TWIST is complementary to all of these methods.

Fully quantized training. FP8 training is now well established (Micikevicius et al., 2022; Peng et al., 2023; DeepSeek-AI, 2024). Recent work targets 4-bit formats, including INT4 (Xi et al., 2023), FP4 (Chmiel et al., 2025), MXFP4 (Tseng et al., 2025; Castro et al., 2025), and NVFP4 (NVIDIA, 2025; Chen et al., 2025b; Panferov et al., 2026; Cook et al., 2025). These recipes combine microscaling formats (Rouhani et al., 2023), Hadamard transforms, stochastic rounding, and selective high-precision layers. TWIST can replace the fixed transform in such recipes: its backward variant plugs into the unbiased estimator of Quartet II (Panferov et al., 2026).

3 METHODOLOGY

3.1 BACKGROUND ON THE WUSH TRANSFORM

Consider a linear layer with weights $W \in \mathbb{R}^{d_{\text{out}} \times d}$ applied to a batch of input activations $X \in \mathbb{R}^{d_{\text{batch}} \times d}$, where d is the number of input channels, d_{out} is the number of output channels, and d_{batch} is the batch size; samples are stacked as rows, and the output is $XW^\top \in \mathbb{R}^{d_{\text{batch}} \times d_{\text{out}}}$. In quantization, we approximate the output as $\widehat{Y} = q_X(X)\, q_W(W)^\top$, where q_X and q_W are the quantization operators for activations and weights, respectively. We introduce pre-quantization transformations using an invertible matrix $T \in \mathbb{R}^{d \times d}$, which gives transformed input activations $\widetilde{X} = XT^{-1}$ and transformed weights $\widetilde{W} = WT^\top$. With these transformations, the unquantized output remains unchanged, since $\widetilde{X}\widetilde{W}^\top = XT^{-1}TW^\top = XW^\top$.

Following WUSH (Chen et al., 2026), we restrict T to be block diagonal, partitioning the d coordinates into blocks B of size g_B, a multiple of the quantization group size, so each group lies in a single block. We write $T_B \in \mathbb{R}^{g_B \times g_B}$ for the diagonal block of T acting on block B, and $W_B = W_{:,B} \in \mathbb{R}^{d_{\text{out}} \times g_B}$ and $X_B = X_{:,B} \in \mathbb{R}^{d_{\text{batch}} \times g_B}$ for the corresponding columns. The full-precision output decomposes as a sum of matrices of rank at most g_B, $XW^\top = \sum_B X_B W_B^\top$. The transformed quantized output decomposes in the same way as $\widehat{Y} = \sum_B q_X(X_B T_B^{-1})\, q_W(W_B T_B^\top)^\top$.

During inference, the weight-side transform $W_B T_B^\top$ can be folded into the stored quantized weights offline, so the only additional online operation is the activation-side transform $X_B T_B^{-1}$, applied independently to each block. The block-diagonal form therefore adds only a modest cost of d/g_B independent $g_B \times g_B$ matrix multiplications, and is compatible with groupwise quantization kernels. In practice, this cost is negligible: Chen et al. (2026) provide CUDA kernels that fuse the per-block transform into activation quantization for MXFP4 matrix multiplication. Reproducing their benchmark on an RTX 5090 (Section B), WUSH runs within 2% of a fused block Hadamard transform on every linear layer of Qwen3-32B, at a 4.9–$5.7\times$ speedup over BF16.

Chen et al. (2026) derive closed-form optimal block transforms T_B which minimize a blockwise approximation of the layer's output error, exactly for floating-point quantizers (under a multiplicative noise model) and up to a constant factor for integers. The same WUSH transform is optimal for both joint weight-activation and weight-only quantization: the expected output error splits exactly into a weight-noise term, an activation-noise term, and an interaction term, and all three are proportional to the same functional $\phi(T)$, which pairs the weight and activation energy of each transformed channel. The error is therefore $c_q\, \phi(T)$ with $c_q = \sigma_W^2 + \sigma_X^2 + \sigma_W^2\sigma_X^2$ (5), so the noise levels rescale the objective without changing its minimizer; weight-only quantization is the case $\sigma_X^2 = 0$. Even then the optimal transform depends on the activations, since each channel's weight error is weighted by its activation energy. The proof of Theorem E.3 shows how the minimum is attained: WUSH whitens

the weights, rotates to the singular basis of $L_W^\top L_X$, and rescales so that the transformed weight and activation moments coincide, $T^\star M_W T^{\star\top} = T^{\star-\top} M_X T^{\star-1} = HSH^\top$; the Hadamard rotation H then spreads this common energy evenly across channels. This attains the lower bound $(\operatorname{tr} S)^2/d$ of Lemma E.1; without the rotation the value is $\operatorname{tr}(S^2)$, up to d times larger. Each block transform T_B is computed from the second moments of the corresponding weight and activation blocks. We first form

$$M_{W,B} = \tfrac{1}{d_{\text{out}}} W_B^\top W_B, \qquad M_{X,B} = \tfrac{1}{d_{\text{batch}}} X_B^\top X_B,$$

both in $\mathbb{R}^{g_B \times g_B}$; these are the diagonal blocks of the moments $M_W = W^\top W$ and $M_X = X^\top X$ of Section E, up to normalizations that rescale T_B but do not change the quantization error. We compute lower-triangular Cholesky factors $L_{W,B}, L_{X,B} \in \mathbb{R}^{g_B \times g_B}$ of their damped versions,

$$L_{W,B} L_{W,B}^\top = M_{W,B} + \lambda g_B^{-1} \operatorname{tr}(M_{W,B})\, I, \qquad L_{X,B} L_{X,B}^\top = M_{X,B} + \lambda g_B^{-1} \operatorname{tr}(M_{X,B})\, I,$$

where the damping ratio $\lambda \geq 0$ ensures positive definiteness when a block is rank-deficient. Next, we take the singular value decomposition $L_{W,B}^\top L_{X,B} = U_B S_B V_B^\top$. Letting $H \in \{\pm g_B^{-1/2}\}^{g_B \times g_B}$ denote the normalized Hadamard matrix, the WUSH transform (Theorem E.3) is given by $T_B = HS_B^{1/2} U_B^\top L_{W,B}^{-1} = HS_B^{-1/2} V_B^\top L_{X,B}^\top$. The transform equalizes scales within each block: ignoring damping, both transformed second moments equal $HS_BH^\top$.

3.2 Our Approach: TWIST

Intuitively, we would "just" like to apply the WUSH transform above during training, in order to minimize QAT error. However, applying a data-adaptive transform *during training* poses several challenges. In PTQ, the model is fixed, so the transform can simply be chosen to minimize quantization error for the current weights and activations. During training, however, this instantaneous objective is not sufficient: the model statistics evolve, so a fixed transform can become stale, while repeatedly replacing the transform can disrupt model convergence. Moreover, when backward matrix multiplications are quantized, the transform must leave the gradient estimator unbiased. TWIST addresses these issues via three different regimes:

1. `FROZEN-TWIST` transforms only the forward-pass product $\widehat{Y}$: it estimates $M_{W,B}$ and $M_{X,B}$ once at the start of training, computes the corresponding WUSH transforms T_B, and keeps them fixed thereafter. It targets adaptation settings such as quantization-aware distillation (QAD) and supervised fine-tuning (QAT-SFT), where it consistently reaches lower training loss than our baselines, with gains that reliably translate into downstream improvements starting at 3 bits.
2. `ONLINE-TWIST` targets pretraining, where weight and activation statistics drift substantially. It recomputes the forward-pass WUSH transforms every τ steps from the master weights and the activation statistics of the preceding step. This is *necessary*: since the activation covariance pools tokens across a sequence, using the current step would break autoregressive training. Transforms are refreshed only during the first fraction ρ of training and kept fixed thereafter (by default, $\tau = 20$ and $\rho = 0.75$).
3. `STOCHASTIC-TWIST` transforms the *backward-pass* products $\nabla_X f = (\nabla_Y f)\, W$ and $\nabla_W f = (\nabla_Y f)^\top X$, where $\nabla_Y f \in \mathbb{R}^{d_{\text{batch}} \times d_{\text{out}}}$ is the upstream gradient of the loss f w.r.t. the layer output $Y = XW^\top$. The WUSH transforms are computed from the statistics of the operands of each product: the quantized transformed weights or activations from the forward pass, and the instantaneous upstream-gradient statistics of the current step. The data-dependent factors are refreshed every $\tau = 20$ steps and never frozen. Importantly, we use a Hadamard transform with fresh random signs at every step.

3.3 Theoretical Guarantees

In Appendix E, we provide analyses of TWIST QAT variants under standard modelling, i.e. with full-precision master weights and straight-through gradients, as a perturbed-iterate SGD method. The model is formalized in Definition E.1.

1. For **linear regression**, we show that the expected quantized loss splits exactly as $J(W,T) = L(W) + c_q\, \phi(W,T)$, where we have the full-precision transform-independent loss $L(W)$ plus a WUSH "functional" at the population activation moment Σ_X (see Section E.5). For fixed weights, choosing T to minimize the QAT objective is therefore exactly the WUSH problem, which Theorem E.3 solves in closed form.

2. For a **fixed transform**, i.e. FROZEN-TWIST, we show that SGD–STE with steps $\eta_t = \Theta(1/t)$ satisfies $\mathbb{E}J(W_K, T) - L(W_*) = \Delta_{\mathrm{FQ}}(T) + \Delta_{\mathrm{STE}}(T) + O(1/K)$ (Theorem E.8). The key observation is that the two residuals do not depend on the step size: Δ_{FQ} is the cost of forward quantization, and Δ_{STE} is the bias of the STE update. Their sum is at most $c_q\,\phi(W_*, T)$, which is minimized by the WUSH transform of $(W_*^\top W_*, \Sigma_X)$. Under weight-only quantization this bound holds with equality, so WUSH minimizes the exact asymptotic loss floor.
3. For **smooth nonlinear models with weight-only quantization**, Theorem E.10 bounds the average squared gradient norm and, under a PL condition, the loss of the quantized model. Each bound consists of standard SGD terms plus a single curvature-weighted distortion term, whose coefficient is independent of the step size. The analysis covers both fixed transforms (FROZEN-TWIST) and transforms refreshed from statistics of earlier batches, matching the refresh rule of ONLINE-TWIST. Under a layerwise curvature bound, this distortion is at most $\sigma_W^2 \sum_B \phi_B(T_B)$, a sum of blockwise WUSH functionals, which the blockwise WUSH transform minimizes (Proposition E.11).

4 Quantized Pretraining with TWIST

Following QuEST (Panferov et al., 2025), we train Llama-like models (Touvron et al., 2023a) with 200M–3B parameters on C4 (Raffel et al., 2020), using additional 50M models for diagnostics and hyperparameter tuning. Unless stated otherwise, we quantize both weights and activations with the QuEST quantizer: a uniform integer grid with one scale per group of 64 values, fitted to the RMS of the group (*Gaussian fitting*). We write W4A4 and W3A3 for 4-bit and 3-bit weights and activations. Transforms act on blocks of 128 input channels. QuEST already applies a Hadamard transform to weights and activations before quantization (*Hadamard preprocessing*), which makes it a natural baseline: we keep the quantizer fixed and only replace the transform, comparing identity, Hadamard, and WUSH. Throughout this section, TWIST denotes ONLINE-TWIST (Section 3), i.e. training with a periodically recomputed and eventually frozen WUSH transform. We report the final loss on the full C4 validation set. Further training details are provided in Appendix C.1.

4.1 Adapting the WUSH transform for pretraining

In PTQ, the WUSH transform is computed once from calibration data. Yet, during pretraining, a transform computed at some point gradually becomes stale. We therefore recompute it periodically: every τ steps, we rebuild the transform from the online weights and activations. Yet, this is not sufficient: Figure 8 shows that, while the run with a periodically recomputed WUSH transform is ahead of Hadamard for most of training, surprisingly, it falls behind towards the end, even though its quantization error is always *lower*! (See Figure 6.)

The basis shift phenomenon. To understand this, we examine the effects of a single recomputation in Table 5. A fresh transform lowers the quantization error of the linear layers, as it should, and early in training the loss follows; late in training, however, the same update raises the loss of the network evaluated on the same master weights and batch. A lower quantization error thus stops translating into a lower loss once the network has adapted to the specific quantized function it is trained with: as the learning rate decays, the model descends into a local minimum of the quantized objective, with later layers compensating for the particular rounding errors of earlier ones, and every change of the transform throws it out of the neighbourhood of that minimum. Figure 9 illustrates: with Hadamard, the fraction of weights that change their grid cell between two evaluations decays with the learning rate to 0.6% at the end of training; with a recomputed WUSH transform, nearly every weight changes its cell after each rebuild.

Going deeper, we note that, as the model optimizes, consecutive WUSH transforms of the same layer are essentially unrelated, behaving nearly as random orthogonal matrices: the distance between steps t and $t+\tau$, $\|T_{t+\tau} - T_t\|_F / \|T_t\|_F$, is 1.38 throughout training, against $\sqrt{2} \approx 1.41$ for random orthogonal matrices. When eigenvalues are close, small changes in the weight and activation second moments can cause large rotations of the eigenvectors. As a result, the basis is recomputed without alignment to the previous one, and these basis changes can therefore alter the grid substantially.

Solution: Freezing the transform. We therefore stop updating the transform at a fraction ρ of training and keep it frozen afterwards. This leaves two TWIST-specific hyperparameters: the update interval τ and the freezing point ρ. Figure 1 sweeps both hyperparameters at 50M ($D = 100N$) and

200M ($D = 25N$) in W3A3. Both matter, and the freezing point matters more: never freezing costs about 40 mNat (milli-nats, i.e. 10^{-3} nat units of validation loss) at 50M and 26 mNat at 200M, while the update interval accounts for up to about 7 mNat. (The run-to-run noise is about 1 mNat.) Freezing at 50–75% of training is best at both sizes, and the loss is flat over a wide range of update intervals. Since the optimum sits at the same place for both sizes, we fix $\tau = 20$ and $\rho = 0.75$ for all further experiments.

Statistics collection. The transform of a layer is rebuilt from the second moments of its weights and input activations, $M_W = W^\top W$ and $M_X = X^\top X = \sum_t x_t x_t^\top$ (Section 3), where x_t is the input activation of token t of the batch. As discussed, M_X cannot be taken from the current batch, so we compute M_X from the activations of the training step preceding the rebuild.

Efficient implementation. The cost of applying the WUSH transform is small relative to the layer's matrix multiplication: the activation and weight transforms require fractions g_B/d_{out} and g_B/d_{batch} of its FLOPs, respectively. With $g_B = 128$, together they account for 5% of the linear-layer FLOPs of a transformer block at 200M and 2.5% at 3B. The additional cost comes from rebuilding the transform every τ steps. On the preceding step, we accumulate the block moments $X_B^\top X_B$ and $W_B^\top W_B$. Accumulating activation moments costs as much as applying the transform. We then average these moments across data-parallel workers using a reduce-scatter, which distributes disjoint subsets of blocks among the workers. Each worker computes the transforms for its blocks using one $g_B \times g_B$ Cholesky factorization and one symmetric eigendecomposition per block. Once these are ready, we gather the transforms across workers with an all-gather. The communicated tensors have size $d \times g_B$ per layer, a fraction g_B/d_{out} of the weight size. A rebuild takes 0.5–0.6 s at 200M–800M. With $\tau = 20$, this adds 5–12% to iteration time before freezing, as shown in Table 4. Once the transform is frozen, this overhead disappears.

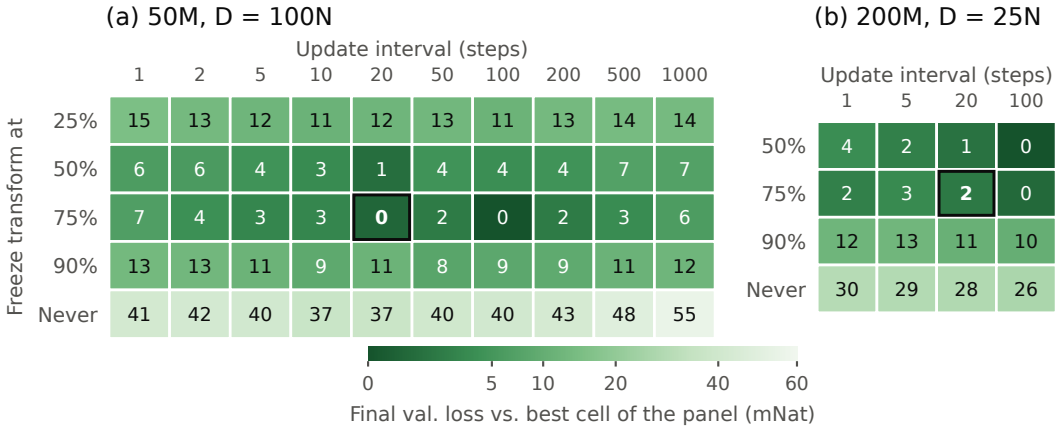

Figure 1: Transform freezing point against transform update interval at W3A3, for (a) a 50M model trained on $D = 100N$ tokens and (b) a 200M model trained on $D = 25N$ tokens. Each cell is the final validation loss minus the best cell of its panel, in mNat (10^{-3} nats); the outlined cell is the operating point used in the rest of the paper. Freezing the transform at 75% of training is worth about 40 mNat over never freezing at 50M, and 26 mNat at 200M.

4.2 Main results for pretraining with TWIST

Table 1 reports the validation loss of the three transforms at W4A4 and W3A3 for model sizes from 200M to 3B, together with the gap to full precision and the share of the Hadamard gap that TWIST closes. At W3A3 the Hadamard transform is unstable with a quantization group of 64, so for this setting we report Hadamard with one scale per row, the better of the two. TWIST improves over Hadamard at every model size and both precisions, closing 15–44% of the Hadamard gap, with each transform at its best-performing group size. Most importantly, the share of the gap that TWIST closes tends to grow with model size, reaching 37% at W4A4 and 44% at W3A3 at 3B, which suggests that the advantage of a data-adaptive transform will only increase at larger scale. We attribute this to the activation distribution: outliers in activations, and more generally structure that a fixed transform cannot adapt to, emerge with model size (Dettmers et al., 2022), leaving a data-adaptive transform more room for improvement in larger models.

Table 1: Identity, Hadamard and TWIST across model sizes from 200M to 3B at INT4 and INT3. TWIST removes 15–44% of the Hadamard gap at every size and both bit widths, and the share it removes appears to grow with model size.

			INT4				INT3			
Size	Method	FP loss	Val. loss	Abs. gap	Rel. gap	Gap red.	Val. loss	Abs. gap	Rel. gap	Gap red.
200M	Identity	2.812	2.901	+0.089	227.4%	−127.4%	2.982	+0.170	160.4%	−60.4%
	Hadamard		2.851	+0.039	100.0%	0.0%	2.918	+0.106	100.0%	0.0%
	TWIST		**2.844**	**+0.032**	**81.9%**	**18.1%**	**2.901**	**+0.089**	**84.0%**	**16.0%**
430M	Identity	2.622	2.708	+0.086	242.7%	−142.7%	2.784	+0.162	175.5%	−75.5%
	Hadamard		2.657	+0.035	100.0%	0.0%	2.714	+0.092	100.0%	0.0%
	TWIST		**2.650**	**+0.028**	**78.9%**	**21.1%**	**2.700**	**+0.078**	**84.8%**	**15.2%**
800M	Identity	2.485	2.560	+0.075	192.9%	−92.9%	2.633	+0.148	148.5%	−48.5%
	Hadamard		2.524	+0.039	100.0%	0.0%	2.585	+0.100	100.0%	0.0%
	TWIST		**2.511**	**+0.026**	**66.2%**	**33.8%**	**2.558**	**+0.072**	**72.6%**	**27.4%**
3B	Hadamard	2.244	2.275	+0.031	100.0%	0.0%	2.350	+0.106	100.0%	0.0%
	TWIST		**2.264**	**+0.020**	**62.7%**	**37.3%**	**2.304**	**+0.059**	**56.3%**	**43.7%**

Token budget. The models above are trained on $D = 25N$ tokens, slightly above the compute-optimal ratio, whereas models meant for deployment are trained far past it, and quantization is known to become harder as the token budget grows (Chen et al., 2025a): the gap of a quantized model to full precision widens with D/N, so a method that helps at $25N$ need not help at $200N$. Figure 2(a, b) varies the budget at 200M from $25N$ to $200N$ tokens. The gap of both transforms to full precision grows slowly with the budget, while the distance stays constant: TWIST closes 8–18% of the Hadamard gap at W4A4 and 12–16% at W3A3, with no trend in the budget.

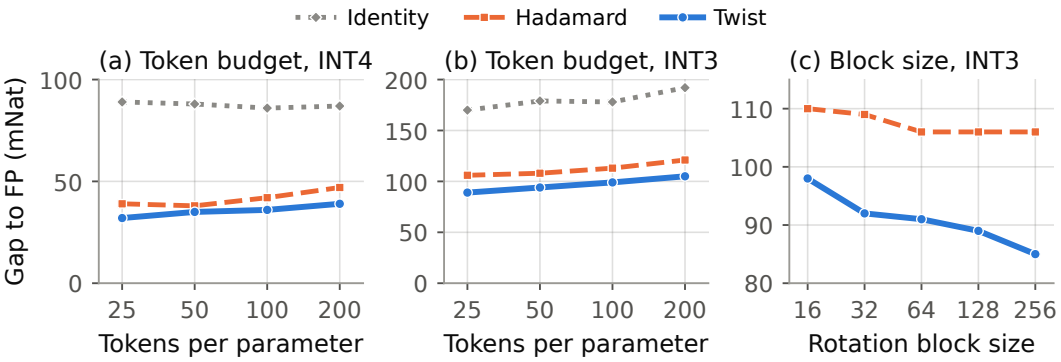

Figure 2: Gap to full precision of the 200M model, in mNat. (a, b) Token budgets from $25N$ to $200N$ at INT4 and INT3. TWIST retains its advantage over a wide range of token budgets, closing 8–18% of the Hadamard gap at INT4 and 12–16% at INT3. (c) Rotation block size at INT3 ($D = 25N$): the TWIST advantage increases with rotation block size. Hadamard at INT3 uses one scale per row, as in Table 1; all other cells use quantization group 64.

Weight-only quantization. TWIST does not improve all quantization settings, however. Weight-only quantization illustrates the limitation discussed in Section 4.1: the network can adapt to a persistent pattern of weight rounding errors, and updating the transform disrupts this adaptation. At 200M, TWIST matches Hadamard at W3A16 but trails it by 10 mNat at W2A16; freezing earlier removes this gap, supporting this explanation. In contrast, activation rounding errors vary across inputs, leaving less opportunity to compensate for a specific error pattern. TWIST's lower quantization error can therefore translate more directly into better loss, giving a 24 mNat improvement over Hadamard at W16A3.

Transform block size. A further check that TWIST behaves as intended is the block size of the transform. A larger block gives the WUSH transform more degrees of freedom to fit the statistics of the layer, whereas the data-independent Hadamard transform has nothing to gain from it. Figure 2(c) confirms this at 200M W3A3: the Hadamard loss is flat across block sizes from 16 to 256, while the TWIST loss keeps improving, by 13 mNat over the same range.

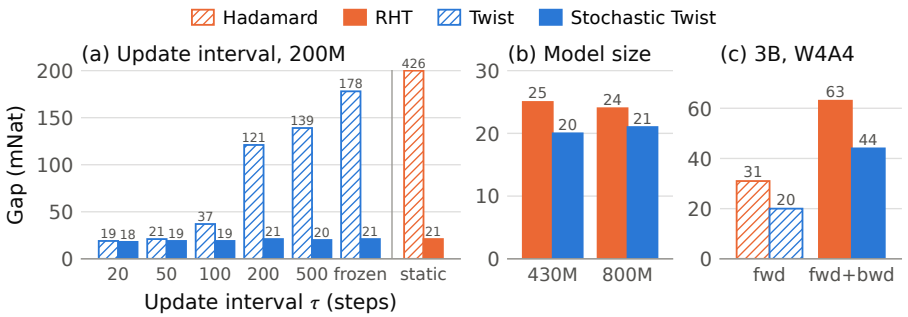

Figure 3: Quantized backward pass, gaps in mNat. (a) Without randomization, a deterministic WUSH transform in the backward pass diverges due to biased gradient estimates. STOCHASTIC-TWIST keeps the gradient estimator unbiased and remains stable across a wide range of refresh intervals. The same holds for plain Hadamard: randomizing it (RHT) helps. (b) STOCHASTIC-TWIST beats the data-agnostic RHT. (c) The improvements carry over to the full pipeline at 3B.

4.3 Quantized backward pass

To investigate the backward pass, we integrate the WUSH transform into the *backward* recipe of Quartet II (Panferov et al., 2026). The WUSH transform (Section 3 and Theorem E.3) factorizes as $T_B^\top = W_B^\top U_B^\top S_B^\top H^\top = \Omega_B^\top H^\top$, i.e. into WUS and H. We retain the data-dependent factor $\Omega_B^\top$ and replace $H^\top$ by a Hadamard transform with fresh random signs at every step. The resulting transform, which we call STOCHASTIC-TWIST, retains the balancing of the WUSH transform but keeps the gradient estimator unbiased. This transform is rebuilt every 20 steps and never frozen.

Figure 3(a) shows why the randomization is needed. With dgrad quantized to INT3 at 200M (group and block size 128, full-precision forward pass), STOCHASTIC-TWIST is barely sensitive to its update interval, whereas the deterministic backward WUSH transform degrades quickly as the update interval grows, and a transform frozen early in training is as harmful as a fixed Hadamard transform: a deterministic transform that stays fixed biases the gradient, and the bias does not average out. That the deterministic transform matches STOCHASTIC-TWIST at $\tau = 20$ we attribute to the mechanism of Section 4.1: every rebuild redraws the basis, which acts as a randomization to some extent.

At 430M and 800M, with dgrad quantized to INT3 and the forward pass in full precision, STOCHASTIC-TWIST reduces the gap to full precision by 3–5 mNat, or 12.5–20%, relative to RHT (Figure 3(b)).

Putting everything together. Finally, we combine ONLINE-TWIST in the forward pass with STOCHASTIC-TWIST in both backward matrix multiplications at 3B, using W4A4 forward and INT3 backward quantization. The improvements of the two passes add up: validation loss drops from 2.307 with Hadamard/RHT to 2.288, narrowing the gap to full precision from 63 to 44 mNat (Figure 3(c)).

5 Model Adaptation Experiments

Following Counathe et al. (2026), we consider 1) *healing* an already trained LLM via quantization-aware distillation (QAD), and 2) *quantized fine-tuning* of a base model to acquire new capabilities.

Shared experimental setup. We evaluate both adaptation scenarios on Qwen3-4B-Thinking-2507 (Yang et al., 2025) and Qwen3-32B for healing, and Qwen3-4B-Base and Qwen3-14B-Base for fine-tuning. We use symmetric absmax quantizers and aggressive compression, using INT4 and INT3 formats with a group size of 128 (and, for fine-tuning, MXFP4) under both weight-only (W-only) and W+A quantization. We compare TWIST against QAT without smoothing (Identity) and QAT with a full Hadamard. The WUSH transform is block-diagonal with the block size matched to the quantization group: 128 for the INT formats and 32 for MXFP4. All runs use AdamW with standard parameters. After warmup, the learning rate is held constant for QAD (Counathe et al., 2026), while QAT decays it linearly to 0. See Appendix D.2 for additional details. We compute the WUSH

transform once, at the start of training, from weight and activation statistics collected on the first 2048 training sequences (see Appendix D.2).

QAD Healing. QAD distills the full-precision teacher into its quantized copy. We train on OpenPerfectBlend (Counathe et al., 2026), an open-source version of Xu et al. (2024). We first measure how closely each student recovers its full-precision teacher, using the forward KL divergence and top-1 token agreement over the response tokens of 128 held-out conversations (Table 6 and Figure 10a in Appendix D.1; Figure 4a shows the training curves at INT3 W+A). TWIST achieves the lowest KL and the highest top-1 agreement in all configurations: relative to Hadamard, it reduces KL by 8–27%, with largest gains under W+A quantization.

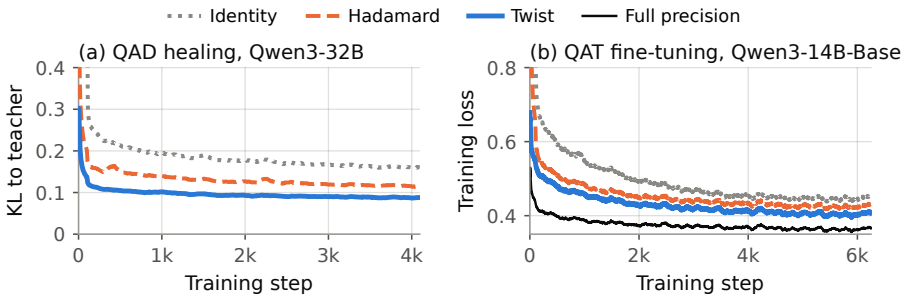

Figure 4: Training curves under joint weight and activation INT3 quantization with group size 128. (a) QAD healing of Qwen3-32B (b) QAT fine-tuning of Qwen3-14B-Base on OpenMathReasoning. Both curves are 100-step moving averages of the per-step values.

We next examine downstream accuracy (last column group of Table 6; Figure 10b), as the mean over nine benchmarks. MMLU, ARC-Challenge, ARC-Easy, HellaSwag, WinoGrande and TruthfulQA are scored zero-shot by log-likelihood with the lm-evaluation-harness (Gao et al., 2023), while GSM8K, MATH-500 and a 500-question subset of MMLU-Pro are generative. At INT4, both smoothing transforms substantially outperform the Identity baseline, but the differences between Hadamard and TWIST fall within evaluation noise. At INT3, TWIST is best in every configuration. We present detailed evaluation details and results in Appendix D.4.

QAT fine-tuning. Next, we fine-tune the 4B and 14B Qwen-3 base models on OpenMathReasoning (Moshkov et al., 2025), containing 306K math problems, paired with chain-of-thought (CoT) solutions by DeepSeek-R1 (DeepSeek-AI, 2025) and QwQ-32B (Qwen-Team, 2025). We keep only CoT solutions to problems with an extracted reference answer, retain a single solution per problem, and finally subsample to 100K training sequences. We evaluate on MATH-500 (Lightman et al., 2023); Figure 4b shows the training curves at INT3 W+A. The 4B base model scores 27.4% and reaches 71.2% after SFT training with no quantization, while the 14B base starts from 49.8% and reaches 76.8%.

We compare training directly with QAT, and fine-tuning in full precision followed by PTQ. Table 7 and Figure 10c (Appendix D.1) show that as quantization becomes more aggressive, QAT becomes preferable to fine-tuning in full precision. Within the QAT variations, TWIST matches or improves over Hadamard in 11 of 12 settings, with its clearest gains under joint weight–activation quantization. At INT3 W+A, TWIST improves MATH-500 accuracy by 2.0 points at 4B and 2.2 points at 14B.

6 Conclusion

We showed, for the first time, that data-adaptive transforms, until now used only in PTQ, can be fitted online during quantization-aware training, efficiently and stably, in both the forward and the backward pass. Our TWIST method consistently improves on the fixed Hadamard rotations used by state-of-the-art QAT recipes. Along the way, we identified a *basis-shift* phenomenon specific to adaptive transforms. Natural next steps include 1) scaling TWIST to larger models and longer token budgets, 2) combining it with microscaling FP4 formats and fused kernels in fully-quantized training pipelines; and 3) designing basis-aware incremental updates that may align successive transforms.

AI USE STATEMENT

In this work, we used generative AI tools for helping develop theoretical models, checking mathematical claims, providing critical assisting in the writing of proofs, implementing methods, paper reformatting, and interpreting results. We have reviewed all AI-assisted work. We checked LLM-generated research ideas for potential plagiarism through a literature survey. LLM-generated code was verified and tested for correctness by the authors. We take full responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Throughout this work, we have strictly adhered to the ICLR Code of Ethics. All datasets utilized in our experiments are publicly available and widely recognized within the scientific community. These datasets do not contain any personally identifiable information or sensitive content. Our work does not involve human subjects, animals, or any form of personal data collection. We have thoroughly considered potential dual-use concerns and do not foresee any harmful applications of our methods. There are no conflicts of interest to declare, and no external sponsorship influenced the outcomes of this research. All experiments were conducted with integrity and transparency.

REPRODUCIBILITY STATEMENT

We are committed to ensuring that our work is transparent and reproducible. To facilitate this, clear explanations of any assumptions and a complete proof of the claims have been included in the main text and appendix. We plan to provide the full implementation of our framework upon the final revision of our paper.

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.

Saleh Ashkboos, Mahdi Nikdan, Soroush Tabesh, Roberto L. Castro, Torsten Hoefler, and Dan Alistarh. HALO: Hadamard-assisted lower-precision optimization for LLMs. *arXiv preprint arXiv:2501.02625*, 2025.

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

Miguel P. Bento and João F. Seabra. GaugeQuant: Online learning of quantization-optimal bases from LLM symmetries. *arXiv preprint arXiv:2607.20757*, 2026.

Roberto L Castro, Andrei Panferov, Soroush Tabesh, Oliver Sieberling, Jiale Chen, Mahdi Nikdan, Saleh Ashkboos, and Dan Alistarh. Quartet: Native FP4 training can be optimal for large language models. *arXiv preprint arXiv:2505.14669*, 2025.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. QuIP: 2-bit quantization of large language models with guarantees. In *Advances in Neural Information Processing Systems*, 2023.

Jiale Chen, Vage Egiazarian, Roberto L. Castro, Torsten Hoefler, and Dan Alistarh. WUSH: Near-optimal adaptive transforms for LLM quantization. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ZsECxUkbKB.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. EfficientQAT: Efficient quantization-aware training for large language models. *arXiv preprint arXiv:2407.11062*, 2024.

Mengzhao Chen, Chaoyi Zhang, Jing Liu, Yutao Zeng, Zeyue Xue, Zhiheng Liu, Yunshui Li, Jin Ma, Jie Huang, Xun Zhou, et al. Scaling law for quantization-aware training. *arXiv preprint arXiv:2505.14302*, 2025a.

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. *arXiv preprint arXiv:2510.27527*, 2025b.

Brian Chmiel, Maxim Fishman, Ron Banner, and Daniel Soudry. FP4 all the way: Fully quantized training of LLMs. *arXiv preprint arXiv:2505.19115*, 2025.

Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. PACT: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018.

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. *arXiv preprint arXiv:2512.02010*, 2025.

Vincent Counathe, Ben Athiwaratkun, Christopher De Sa, and Tianyi Zhang. Quasar: Lowering the loss floor of quantization-aware training with loss-aware reconstruction, 2026. URL https://arxiv.org/abs/2608.13966.

Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. BinaryConnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems*, 2015.

DeepSeek-AI. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. *Advances in neural information processing systems*, 35: 30318–30332, 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.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023. URL https://zenodo.org/records/10256836.

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. In *International Conference on Learning Representations*, 2025.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *Advances in Neural Information Processing Systems*, 2016.

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Houyi Li, Wenzhen Zheng, Qiufeng Wang, Hanshan Zhang, Zili Wang, Shijie Xuyang, Yuantao Fan, Zhenyu Ding, Haoying Wang, Ning Ding, et al. Predictable scale: Part i, step law–optimal hyperparameter scaling law in large language model pretraining. *arXiv preprint arXiv:2503.04715*, 2025.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

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. In *Advances in Neural Information Processing Systems*, 2024a.

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 on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, 2024b.

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.

I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit LLMs: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*, 2024a.

Yuexiao Ma, Huixia Li, Xiawu Zheng, Feng Ling, Xuefeng Xiao, Rui Wang, Shilei Wen, Fei Chao, and Rongrong Ji. AffineQuant: Affine transformation quantization for large language models. In *International Conference on Learning Representations*, 2024b.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

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

Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset, 2025. URL https://arxiv.org/abs/2504.16891.

Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. Overcoming oscillations in quantization-aware training. In *International Conference on Machine Learning*, 2022.

NVIDIA. NVIDIA blackwell architecture technical brief, 2024. URL https://resources.nvidia.com/en-us-blackwell-architecture.

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

Andrei Panferov, Jiale Chen, Soroush Tabesh, Roberto L Castro, Mahdi Nikdan, and Dan Alistarh. QuEST: Stable training of LLMs with 1-bit weights and activations. *arXiv preprint arXiv:2502.05003*, 2025.

Andrei Panferov, Erik Schultheis, Soroush Tabesh, and Dan Alistarh. Quartet II: Accurate LLM pretraining in NVFP4 by improved unbiased gradient estimation. *arXiv preprint arXiv:2601.22813*, 2026.

Houwen Peng, Kan Wu, Yixuan Wei, Guoshuai Zhao, Yuxiang Yang, Ze Liu, Yifan Xiong, Ziyue Yang, Bolin Ni, Jingcheng Hu, Ruihang Li, Miaosen Zhang, Chen Li, Jia Ning, Ruizhe Wang, Zheng Zhang, Shuguang Liu, Joe Chau, Han Hu, and Peng Cheng. FP8-LM: Training FP8 large language models. *arXiv preprint arXiv:2310.18313*, 2023.

Qwen-Team. Qwq-32b: Embracing the power of reinforcement learning, March 2025. URL https://qwenlm.github.io/blog/qwq-32b/.

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.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. XNOR-Net: ImageNet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, 2016.

Bita Darvish Rouhani, Ritchie Zhao, Ankit More, Mathew Hall, Alireza Khodamoradi, Summer Deng, Dhruv Choudhary, Marius Cornea, Eric Dellinger, Kristof Denolf, Dusan Stosic, 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.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. OmniQuant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations*, 2024.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. Massive activations in large language models. In *Conference on Language Modeling*, 2024a.

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. *arXiv preprint arXiv:2410.09426*, 2024b.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP#: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *International Conference on Machine Learning*, 2024.

Albert Tseng, Tao Yu, and Youngsuk Park. Training LLMs with MXFP4. *arXiv preprint arXiv:2502.20586*, 2025.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. BitNet: Scaling 1-bit transformers for large language models. *arXiv preprint arXiv:2310.11453*, 2023.

Hongyu Wang, Shuming Ma, and Furu Wei. BitNet v2: Native 4-bit activations with hadamard transformation for 1-bit LLMs. *arXiv preprint arXiv:2504.18415*, 2025.

Quan Wei, Chung-Yiu Yau, Hoi-To Wai, Yang Katie Zhao, Dongyeop Kang, Youngsuk Park, and Mingyi Hong. RoSTE: An efficient quantization-aware supervised fine-tuning approach for large language models. *arXiv preprint arXiv:2502.09003*, 2025.

Haocheng Xi, Changhao Li, Jianfei Chen, and Jun Zhu. Training transformers with 4-bit integers. In *Advances in Neural Information Processing Systems*, 2023.

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

Meng Xin, Sweta Priyadarshi, Jingyu Xin, Bilal Kartal, Aditya Vavre, Asma Kuriparambil Thekkumpate, Zijia Chen, Ameya Sunil Mahabaleshwarkar, Ido Shahaf, Akhiad Bercovich, Kinjal Patel, Suguna Varshini Velury, Chenjie Luo, Zhiyu Cheng, Jenny Chen, Chen-Han Yu, Wei Ping, Oleg Rybakov, Nima Tajbakhsh, Oluwatobi Olabiyi, Dusan Stosic, Di Wu, Song Han, Eric Chung, Sharath Turuvekere Sreenivas, Bryan Catanzaro, Yoshi Suhara, Tijmen Blankevoort, and Huizi Mao. Quantization-aware distillation for nvfp4 inference accuracy recovery, 2026. URL https://arxiv.org/abs/2601.20088.

Tengyu Xu, Eryk Helenowski, Karthik Abinav Sankararaman, Di Jin, Kaiyan Peng, Eric Han, Shaoliang Nie, Chen Zhu, Hejia Zhang, Wenxuan Zhou, Zhouhao Zeng, Yun He, Karishma Mandyam, Arya Talabzadeh, Madian Khabsa, Gabriel Cohen, Yuandong Tian, Hao Ma, Sinong Wang, and Han Fang. The perfect blend: Redefining rlhf with mixture of judges, 2024. URL https://arxiv.org/abs/2409.20370.

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.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.

Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.

CONTENTS

1 Introduction **1**

2 Related work **2**

3 Methodology **3**

3.1 Background on the WUSH Transform 3

3.2 Our Approach: TWIST 4

3.3 Theoretical Guarantees 4

4 Quantized Pretraining with TWIST **5**

4.1 Adapting the WUSH transform for pretraining 5

4.2 Main results for pretraining with TWIST 6

4.3 Quantized backward pass 8

5 Model Adaptation Experiments **8**

6 Conclusion **9**

AI Use Statement **10**

Ethics Statement **10**

Reproducibility Statement **10**

A Notation **15**

B Inference cost of the WUSH transform **18**

C Additional results for quantized pretraining **19**

C.1 Hyperparameters and cost 19

C.2 Quantization error of WUSH and Hadamard 19

D Experimental details - quantized training for adaptation **23**

D.1 Adaptation results 23

D.2 Training and evaluation details 23

D.3 Learning rate sweep 25

D.4 QAD downstream performance 25

D.5 Does the frozen WUSH transform become stale? 28

E Theoretical Analysis **29**

E.1 WUSH: optimal adaptive transforms for joint quantization 29

E.2 Lower bound for the quantization error 30

E.3 Construction of optimal transforms (WUSH) 31

E.4 QAT for linear regression with a quantized forward pass 32

E.5 Risk decomposition and the role of WUSH 32

E.6 Fixed-transform convergence of SGD–STE 33

E.7 Nonlinear QAT with WUSH weight transforms 36

E.8 Weight-only QAT and convergence 36

E.9 Layer structure and the WUSH optimum 38

A Notation

Table 2 collects the notation used throughout the paper. It follows the theoretical analysis of Section E; the main text and the other appendices use the same symbols.

Table 2: Notation.

Symbol	Meaning
Linear layers and transforms	
$d,\ d_{\text{out}},\ d_{\text{batch}}$	Number of input channels, output channels, and samples in a batch
$W \in \mathbb{R}^{d_{\text{out}} \times d}$	Weights of a linear layer (master weights during training)
$X \in \mathbb{R}^{d_{\text{batch}} \times d}$	Batch of input activations, samples stacked as rows
$XW^\top$	Full-precision layer output; Y denotes the layer output, or the regression responses in Section E.4
$q_X,\ q_W$	Activation and weight quantizers

Continued on next page.

Table 2, continued.

Symbol	Meaning
$T \in \mathbb{R}^{d\times d}$	Invertible pre-quantization transform; $T^{-\top} \coloneqq (T^{-1})^\top$
$\widetilde{X} = XT^{-1},\ \widetilde{W} = WT^\top$	Transformed operands, $\widetilde{X}\widetilde{W}^\top = XW^\top$
$\widehat{Y} = q_X(\widetilde{X})\, q_W(\widetilde{W})^\top$	Jointly quantized layer output
$\Delta_X,\ \Delta_W$	Quantization errors of the transformed operands
$B,\ g_B$	Block of consecutive input coordinates and its size
$W_B,\ X_B,\ T_B$	Columns $W_{:,B}$, $X_{:,B}$ of a block, and the diagonal block of T acting on it
$w_{B,j} = W_{j,B}^\top \in \mathbb{R}^{g_B}$	Weight segment of output row j in block B
The WUSH transform	
$M_W,\ M_X$	Weight and activation second-moment matrices, $W^\top W$ and $X^\top X$
$M_{W,B},\ M_{X,B}$	Normalized block moments used in practice (Section 3)
$L_W,\ L_X$	Square factors (e.g. Cholesky), $M_W = L_W L_W^\top$, $M_X = L_X L_X^\top$; blockwise $L_{W,B}, L_{X,B}$
$USV^\top$	Singular value decomposition of $L_W^\top L_X$; blockwise $U_B S_B V_B^\top$
H	Normalized Hadamard matrix; P a general orthogonal equalizer (Remark E.4)
$T^\star = HS^{1/2}U^\top L_W^{-1}$	WUSH transform; blockwise $T_B = H\Omega_B$
$\Omega_B = S_B^{1/2} U_B^\top L_{W,B}^{-1}$	Data-dependent balancing factor of the block transform
$\Gamma = M_W^{1/2} M_X M_W^{1/2}$	Matrix whose eigenvalues are the squared singular values of $L_W^\top L_X$
$\phi(W,T,X),\ \phi(W,T),\ \phi(T)$	WUSH trace functional on a batch, in expectation, and at fixed (W,X)
λ	Damping of the moment matrices
$\mathrm{Diag}(\cdot)$	Keeps the diagonal of a square matrix and zeroes the rest
$\odot,\ \|\cdot\|_{\mathrm{F}},\ \|\cdot\|_*$	Entrywise product, Frobenius norm, nuclear norm
$\|U\|_N$	Weighted norm $\|UN^{1/2}\|_{\mathrm{F}}$ for $N \succ 0$ (matrices) or $\sqrt{v^\top N v}$ (vectors)
Quantization noise model	
$\Xi_X,\ \Xi_W$	Relative (multiplicative) quantization noise, $q_A(M) = M + \Xi_A \odot M$
$\sigma_X^2,\ \sigma_W^2$	Relative noise variances of the activation and weight quantizers
$c_q = \sigma_W^2 + \sigma_X^2 + \sigma_W^2\sigma_X^2$	Joint quantization constant
$\mathbb{E}_q$	Expectation over quantization noise only
QAT for linear regression	
$(x,y) \sim \mathcal{D}$	Sample with input $x \in \mathbb{R}^d$ and response $y \in \mathbb{R}^{d_{\mathrm{out}}}$
$W_*,\ \varepsilon,\ \sigma^2$	Data-generating weights, observation noise, and its scaled variance
$\Sigma_X = \mathbb{E}[X^\top X]$	Population activation moment
$J(W,T)$	Expected loss of the quantized forward pass
$L(W),\ R(W,T)$	Full-precision loss and reconstruction risk, $J = L + R$
$t,\ K,\ \eta_t$	Training step, total number of steps, step size
G_t	SGD–STE update direction at step t
$P_T,\ H_T,\ H_{\mathrm{s}}$	Transform-dependent curvature matrices of the objective and of the mean STE update
$W_T,\ W_{\mathrm{s}}$	Minimizer of $J(\cdot,T)$ and fixed point of the mean STE update
$\Delta_{\mathrm{FQ}}(T),\ \Delta_{\mathrm{STE}}(T)$	Residual losses from forward quantization and from the STE bias
Nonlinear weight-only QAT	
$\theta \in \mathbb{R}^p$	All master weight parameters
$f,\ f^\star$	Full-precision population loss and its infimum
$\mathcal{T},\ \mathcal{T}_t$	Global block-diagonal weight transform (at step t)
$Q_{\mathcal{T}}(\theta) = \mathcal{T}^{-1} q_W(\mathcal{T}\theta)$	Quantized weights
$\mathcal{H},\ \beta = \lambda_{\max}(\mathcal{H})$	Matrix smoothness bound and its largest eigenvalue
$\sigma_g^2,\ \mu$	Stochastic gradient variance and Polyak–Łojasiewicz constant
$e_t,\ D_t$	Weight quantization error and its curvature-weighted distortion
$a_{B,j},\ C_B$	Output-sensitivity factors and input-geometry matrix of a block
$M_{B,t},\ \phi_{B,t}$	Weighted block weight moment and its WUSH functional
$\mathcal{U}_t,\ \mathcal{U}_t^\star$	Layerwise distortion bound and its minimum over transforms
Training schedule and experiments	
τ	Transform refresh interval, in training steps
ρ	Fraction of training after which the transform is frozen
$N,\ D$	Number of non-embedding parameters and of training tokens
(β_1, β_2)	AdamW moment coefficients

Continued on next page.

Table 2, continued.

Symbol	Meaning
WbAb	b-bit weights and activations (e.g. W4A4)
mNat	Milli-nat, 10^{-3} nats; unit for differences in validation loss

B INFERENCE COST OF THE WUSH TRANSFORM

We reproduce the kernel benchmark of Chen et al. (2026) on an RTX 5090, using their MXFP4 kernels, which fuse the block transform into activation quantization. Figure 5 reports the per-layer speedup over BF16.

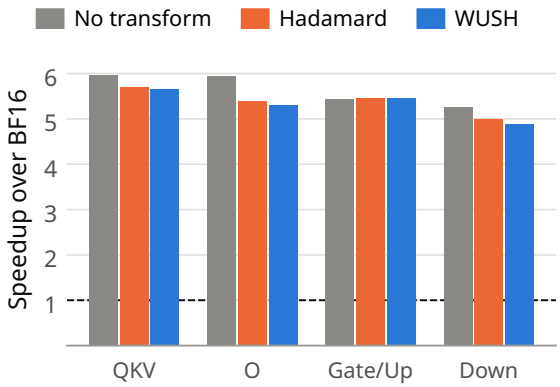

Figure 5: Per-layer speedup over BF16 on an RTX 5090 of the four Qwen3-32B linear layers (fused QKV, O, fused Gate+Up, Down) at M=1024 tokens; every bar is activation quantization plus the same MXFP4 matmul (group size 32) on the FP4 tensor cores, and only the quantizer differs: plain quantization without a transform, the fused block Hadamard, or the fused per-block WUSH. WUSH reaches 4.9–5.7, within 2% of the Hadamard transform on every layer: the per-block matrices cost no more than a shared one once the transform is fused into the quantization kernel.

C Additional results for quantized pretraining

C.1 Hyperparameters and cost

Training Setup. Our pre-training setup follows QuEST (Panferov et al., 2025). We train Llama-like models (Touvron et al., 2023a) with RMSNorm (Zhang & Sennrich, 2019) and SwiGLU (Shazeer, 2020), ranging from 200M to 3B parameters (counting only non-embedding parameters), on the C4 dataset (Raffel et al., 2020) with the Llama 2 tokenizer (Touvron et al., 2023b) and sequence length 512. We additionally use 50M models for diagnostics and hyperparameter tuning. Unless stated otherwise, each model is trained on $D = 25N$ tokens, where N is the number of non-embedding parameters. We use AdamW (Kingma, 2014; Loshchilov, 2017) with $(\beta_1, \beta_2) = (0.9, 0.95)$, decoupled weight decay 0.1 and gradient clipping at 1.0. The learning rate follows a cosine schedule with 10% linear warmup and decays to zero. Batch sizes and peak learning rates are set per model size and token budget following the scaling rule of Li et al. (2025); the exact values are listed in Table 3. Master weights are kept in float32 and the forward and backward passes use bf16 mixed precision (Micikevicius et al., 2017).

Quantization. We use the *trust gradient estimator* of QuEST (Panferov et al., 2025) in the backward pass. All linear layers inside the transformer blocks are quantized; the embeddings and the output projection stay in full precision.

Table 3: Model and training hyperparameters for the quantized-pretraining experiments. Token budgets follow $D = 25N$ with N the nominal size label; the head dimension is 128 and the embeddings are untied. We use AdamW with $\beta = (0.9, 0.95)$, weight decay 0.1, gradient clipping 1.0, a cosine schedule with 10% warmup that starts at $\eta_{\max}/100$ and decays to 0.0, sequence length 512, the Llama-2 tokenizer (32k vocabulary), and bf16 autocast with fp32 master weights.

									Global batch		
Size	Layers	d_{model}	Heads	MLP hidden	Non-emb. params (M)	D (B)	Iterations	Warmup	seq.	tokens	Peak LR
50M	7	768	6	2,048	49.56	1.25	12,715	1,271	192	98,304	4.00×10^{-3}
200M	10	1,280	10	3,584	203.19	5.00	25,431	2,543	384	196,608	2.00×10^{-3}
430M	13	1,664	13	4,608	443.07	10.75	41,007	4,100	512	262,144	1.41×10^{-3}
800M	16	2,048	16	5,632	822.15	20.00	50,862	5,086	768	393,216	1.00×10^{-3}
3B	24	3,072	24	8,192	2,718.06	67.95	86,404	8,640	1,536	786,432	7.07×10^{-4}

Table 4: Iteration time (s) of the W3A3 cells of the main ladder with TWIST, before and after the transform is frozen at 75% of training, and the resulting overhead of the periodic rebuild; the last column is the wall-clock time of one rebuild of all transforms. After the freeze the iteration time matches the Hadamard cell of the same size to within the run-to-run spread.

Size	GPUs	Before freeze	After freeze	Overhead	One rebuild (s)
200M	8	0.275	0.245	$+12.5\%$	0.57
430M	16	0.343	0.315	$+9.1\%$	0.50
800M	16	0.707	0.676	$+4.7\%$	0.62

C.2 Quantization error of WUSH and Hadamard

The weight gradient deserves a separate remark. The forward pass and dgrad reduce over the input and output features of a layer, where the correlations that WUSH exploits are well known; wgrad reduces over the token axis, along which different tokens could be expected to be independent, leaving nothing for a data-adaptive transform to use. Our preliminary experiments nevertheless showed an about 10% lower quantization error of the wgrad product with STOCHASTIC-TWIST than with RHT (Figure 7). We attribute this to the tokens being dependent after all: the input activations of every linear layer are correlated between neighbouring tokens, with a lag-one autocorrelation of 0.61 for the attention output projection, and the correlation does not decay to zero at larger lags. In a control with only wgrad quantized this lower error did not translate into a lower final loss, but the penalty of wgrad quantization is small to begin with (8 mNat at 200M and INT3), so there is little to gain; we keep the data-adaptive transform on wgrad as its advantage may matter at larger scale, where the backward pass contributes more to the overall degradation (Chen et al., 2025a).

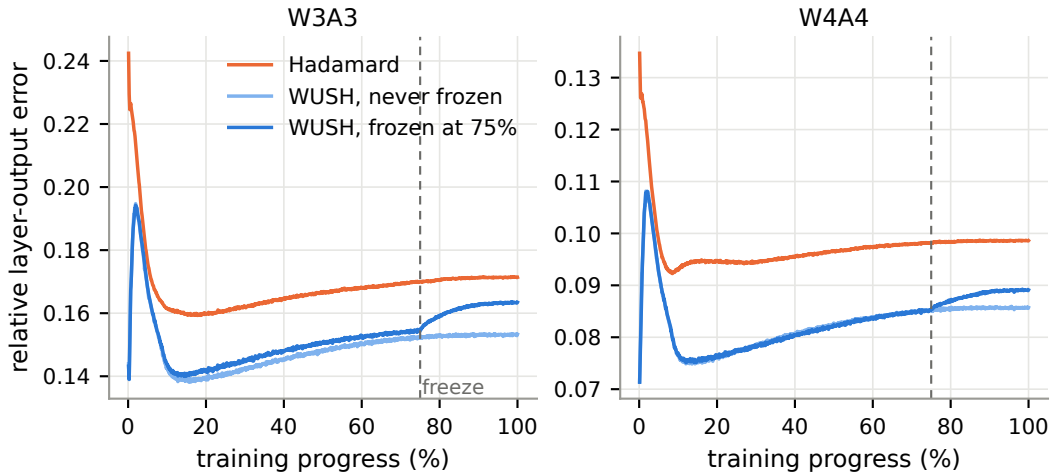

Figure 6: Relative layer-output error of the Hadamard and WUSH transforms over training, 200M, measured on the same weights and batches: WUSH with the transform frozen at 75% of training (the recipe of the paper) and WUSH with the transform recomputed until the end.

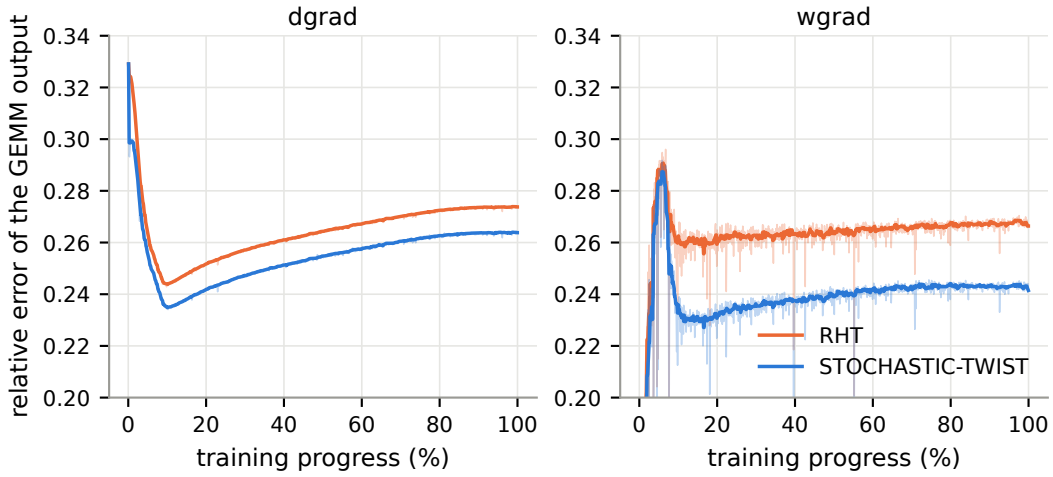

Figure 7: Relative error of the two backward products at INT3 with RHT and STOCHASTIC-TWIST, 200M, measured on the same weights and batches over training.

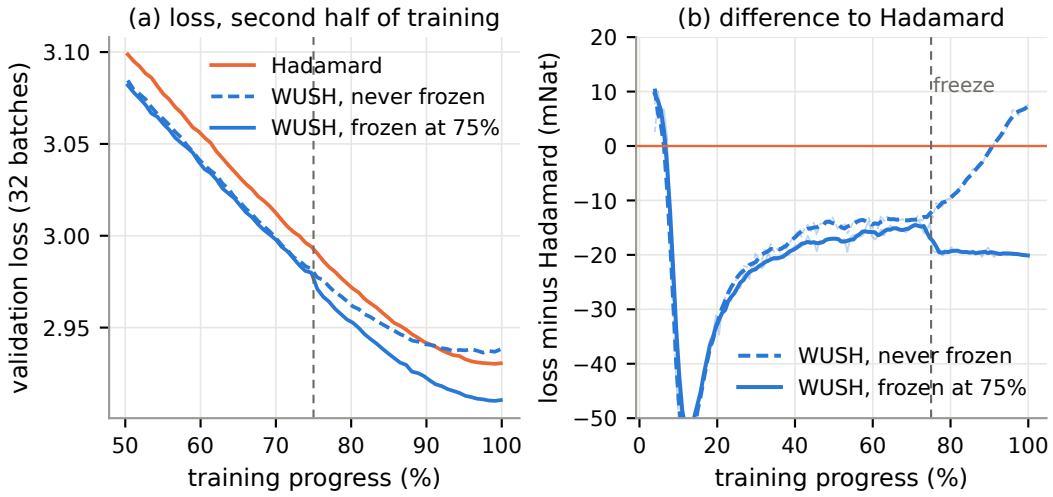

Figure 8: A periodically recomputed WUSH transform against Hadamard over training, 200M W3A3. (a) Validation loss over the second half of training; (b) difference to Hadamard over the whole run. Without freezing, the recomputed transform is ahead by 10–30 mNat for most of training and falls behind at about 90%; frozen at 75%, it keeps its lead.

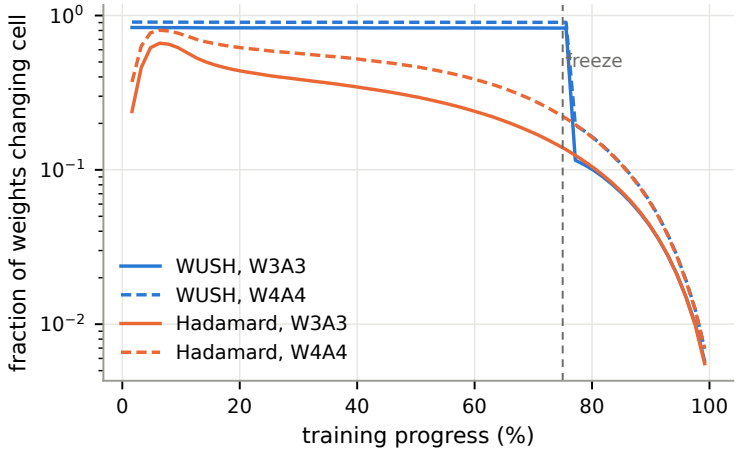

Figure 9: Fraction of quantized weights whose grid cell changed between two consecutive evaluations, 50M. With a fixed transform the fraction decays with the learning rate; with a recomputed WUSH transform it stays at the level of independent draws until the transform is frozen, after which the curve joins the Hadamard one.

Table 5 isolates the effect of a single recomputation of the transform, on the 200M W4A4 cell of the main ladder (QuEST, group 64, block 128, transform rebuilt every 20 steps). The run is resumed from its checkpoints at 10%, 50% and 90% of training, and at the next scheduled rebuild the old and the new transforms are compared on the same master weights. The layer-output error is the error of the quantized output of a linear layer against its full-precision output, on inputs cached from the network with the old transform; the table reports the relative change new/old, median over the 40 quantized layers. The loss is measured on $65{,}536$ held-out C4 tokens with all transforms replaced at once, $\pm$ the standard error over 32 batches; each row is the mean over two consecutive rebuilds.

Table 5: Effect of a single recomputation of the transform at three points of training, 200M W4A4: change of the layer-output error and of the loss.

Point in training	Δ layer-output error	Δ loss (mNat)
10%	-1.1%	-1 ± 1
50%	-0.7%	-1 ± 1
90%	-0.1%	$+2 \pm 1$

D EXPERIMENTAL DETAILS - QUANTIZED TRAINING FOR ADAPTATION

D.1 ADAPTATION RESULTS

Tables 6 and 7 report the full results of the model adaptation experiments discussed in Section 5, and Figure 10 plots them.

Table 6: QAD healing: teacher–student fidelity and downstream accuracy. KL is the held-out divergence to the full-precision teacher and Top-1 its token agreement; downstream evaluation is the mean accuracy over nine benchmarks (see Section 5), for which the FP teacher scores 67.6 (Qwen3-4B) and 75.5 (Qwen3-32B).

			KL ↓			Top-1 ↑			Downstream evaluation ↑		
Model	Quant.	Fmt.	Identity	Hadamard	TWIST	Identity	Hadamard	TWIST	Identity	Hadamard	TWIST
Qwen3-4B	W-only	INT4	0.0263	0.0221	**0.0187**	95.4	95.8	**96.0**	66.0	**67.3**	66.8
		INT3	0.0956	0.0884	**0.0767**	91.5	91.6	**92.2**	57.1	60.3	**62.1**
	W+A	INT4	0.0999	0.0535	**0.0402**	90.8	93.2	**94.2**	59.0	**66.3**	65.5
		INT3	0.3385	0.2200	**0.1853**	83.2	86.4	**87.6**	32.3	41.5	**48.4**
Qwen3-32B	W-only	INT4	0.0163	0.0129	**0.0112**	96.1	96.6	**96.9**	75.3	75.5	**75.9**
		INT3	0.0500	0.0414	**0.0380**	93.7	94.2	**94.4**	73.4	73.7	**74.3**
	W+A	INT4	0.0542	0.0319	**0.0234**	93.2	94.5	**95.6**	74.0	75.2	**75.3**
		INT3	0.1522	0.1081	**0.0834**	88.9	90.5	**91.4**	62.3	67.4	**70.8**

Table 7: MATH-500 accuracy of quantized models adapted from Qwen3-4B-Base and Qwen3-14B-Base for mathematical reasoning. The BF16 fine-tunes score 71.2% (Qwen3-4B) and 76.8% (Qwen3-14B).

		Qwen3-4B				Qwen3-14B			
		QAT			PTQ	QAT			PTQ
Quant.	Fmt.	Identity	Hadamard	TWIST	WUSH RTN	Identity	Hadamard	TWIST	WUSH RTN
W-only	MXFP4	69.6	**70.2**	**70.2**	68.4	76.0	75.6	**77.0**	75.6
	INT4	70.2	71.4	**72.2**	69.0	74.8	**77.0**	75.8	76.2
	INT3	63.8	**66.0**	**66.0**	53.8	71.8	74.6	**74.8**	69.2
W+A	MXFP4	67.2	**68.2**	**68.2**	67.4	74.2	73.4	76.0	**76.6**
	INT4	66.4	66.6	68.8	**69.2**	71.8	75.4	**76.6**	76.0
	INT3	41.8	53.8	**55.8**	15.2	60.0	67.4	**69.6**	51.6

D.2 TRAINING AND EVALUATION DETAILS

Calibration. The WUSH transform is computed once, at the start of training, from weight and activation statistics collected on the first 2048 training sequences. Because training samples vary in length, we exclude padding tokens when accumulating activation statistics, and we sort sequences by length so that sequences of similar length are batched together, reducing wasted computation on padding. Calibration is parallelized across data-parallel workers, aggregating the per-worker activation Gram matrices with an all-reduce. For QAT fine-tuning, the transform is calibrated on the base model, unlike QAD, where it is fitted to a model that already solves the task.

QAD healing. From Open-PerfectBlend, we keep the conversations that contain both a user and an assistant turn, hold out the last 1% of the training split for the fidelity evaluation, and truncate sequences to 4096 tokens. Qwen3-4B-Thinking-2507 is distilled with its (thinking) chat template; Qwen3-32B, a hybrid thinking/non-thinking model, is distilled and evaluated in non-thinking mode. Every student trains for 4096 steps at a global batch of 32 sequences for Qwen3-4B and 128 sequences for Qwen3-32B. For the generative downstream benchmarks, the thinking model (Qwen3-4B) is sampled with Qwen's recommended settings (temperature 0.6, top-p 0.95, top-k 20), while Qwen3-32B is decoded greedily in non-thinking mode.

QAT fine-tuning. Each OpenMathReasoning sequence is rendered with the Qwen3 chat template, with the chain-of-thought trace as the assistant's thinking block. On MATH-500, we sample one

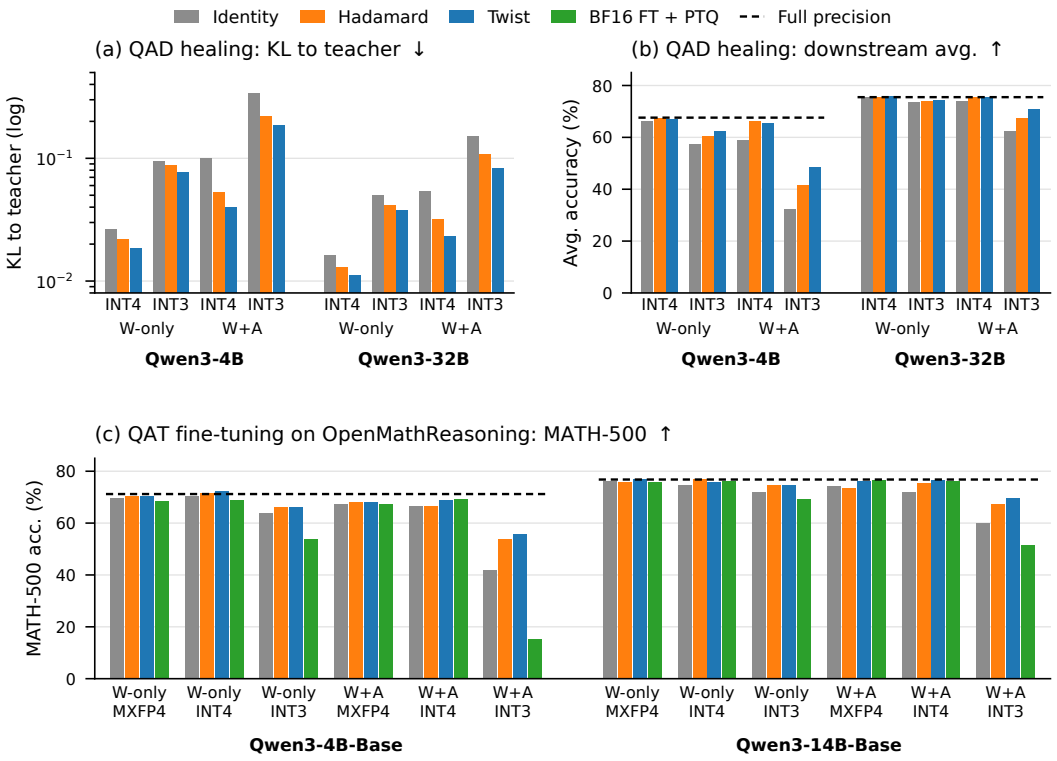

Figure 10: Model adaptation results. (a, b) QAD healing of Qwen3-4B-Thinking-2507 and Qwen3-32B: KL divergence to the full-precision teacher (log scale) and mean downstream accuracy over nine benchmarks (Table 6). (c) QAT fine-tuning of Qwen3-4B-Base and Qwen3-14B-Base on OpenMathReasoning, evaluated on MATH-500, against BF16 fine-tuning followed by PTQ (Table 7). Dashed lines mark the full-precision teacher in (b) and the BF16 fine-tune in (c).

solution per problem through the chat template in thinking mode (temperature 0.6, top-p 0.95, top-k 20). For the PTQ baseline, we quantize the full-precision fine-tune in the same format with the WUSH transform and round-to-nearest. The selected learning rates for every configuration are listed in Section D.3.

D.3 Learning rate sweep

Tables 8 and 9 list the selected learning rates (lowest loss) for every configuration in Tables 6 and 7.

Table 8: QAD healing: selected learning rates ($\times 10^{-5}$).

		Qwen3-4B			Qwen3-32B		
Quant.	Fmt.	Identity	Hadamard	TWIST	Identity	Hadamard	TWIST
W-only	INT4	2	2	1	2	2	1
	INT3	6	3	3	8	4	4
W+A	INT4	4	2	1	4	2	1
	INT3	6	6	3	8	8	4

Table 9: QAT fine-tuning: selected learning rates ($\times 10^{-5}$). The full-precision fine-tune (the PTQ source) uses 10^{-4} for Qwen3-4B and 5×10^{-5} for Qwen3-14B.

		Qwen3-4B			Qwen3-14B		
Quant.	Fmt.	Identity	Hadamard	TWIST	Identity	Hadamard	TWIST
W-only	MXFP4	10	10	10	5	10	5
	INT4	10	10	10	10	10	5
	INT3	10	10	10	10	10	10
W+A	MXFP4	10	10	10	10	10	5
	INT4	10	10	10	10	10	5
	INT3	20	20	20	20	10	10

D.4 QAD downstream performance

Figures 11 and 12 show per-task downstream accuracy after QAD healing, for Qwen3-4B and Qwen3-32B, under weight-only (W-only) and joint weight–activation (W+A) quantization at INT4 and INT3.

At INT4, every transform on both models recovers the teacher to within about 2 pt. The one exception is vanilla QAD on the harsher W+A setting for the smaller model, which loses 8.6 pt, mostly on GSM8K and MMLU-Pro. INT3 is where the transforms differ. TWIST has the highest average in all four INT3 panels, and the gap grows as the setting gets harder: smaller models, joint quantization.

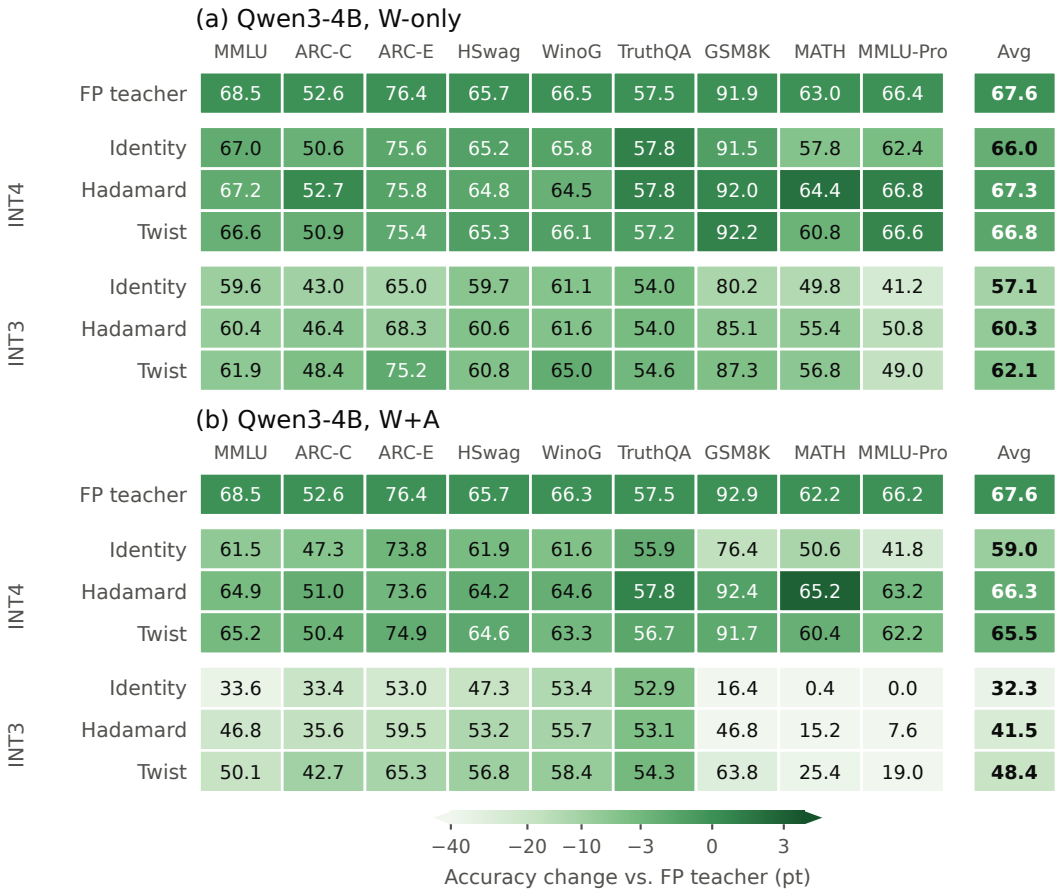

Figure 11: Per-task downstream accuracy of Qwen3-4B after QAD healing

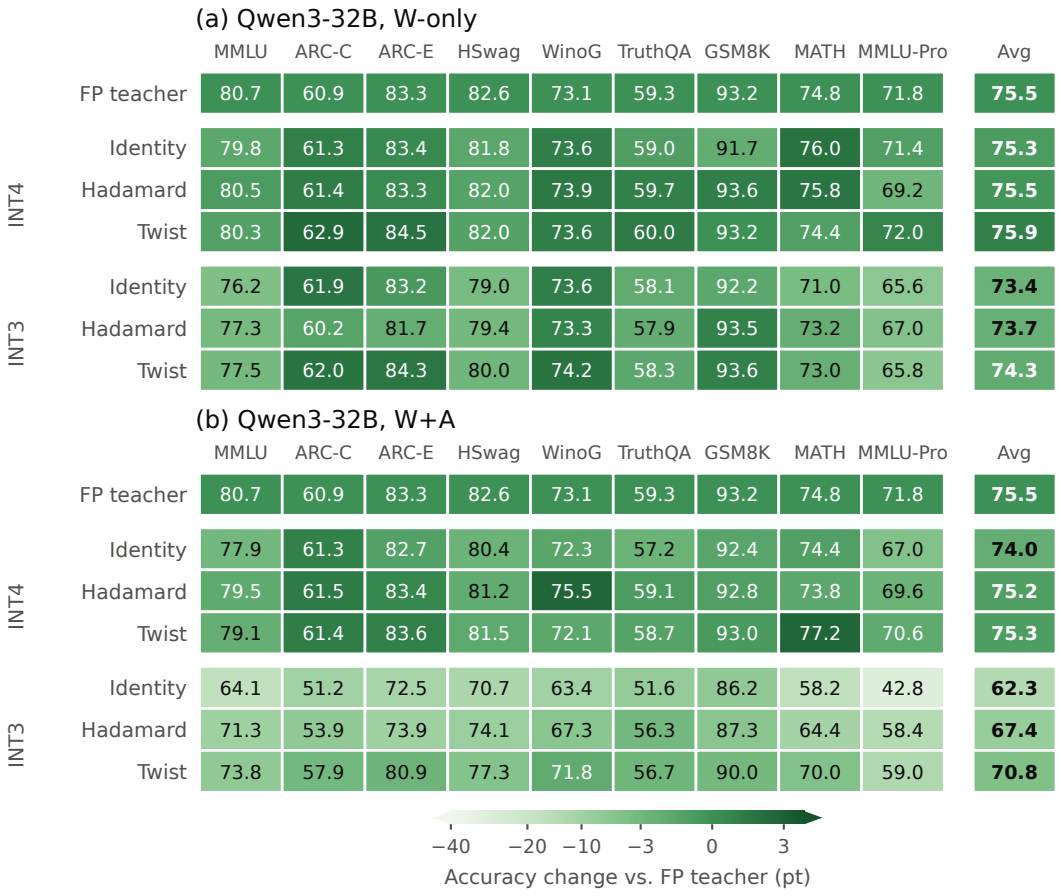

Figure 12: Per-task downstream accuracy of Qwen3-32B after QAD healing

D.5 Does the frozen WUSH transform become stale?

As in late-stage pretraining, we freeze the WUSH transform for both QAD and QAT-SFT: we found that every refresh introduces a small loss bump that is not fully recovered during adaptation. We evaluate the staleness of the frozen transform at the end of training. Since the WUSH transform is not uniquely determined, we compare transforms through their effect on the output error of quantized layers. Concretely, we refit WUSH on the final weights using the original calibration sequences and measure per-layer output MSE on held-out data (roughly 1M tokens per setting). As Figure 13 shows, the frozen transform remains close to a freshly computed one: its median MSE over layers is within 1.01–1.11× that of the fresh transform.

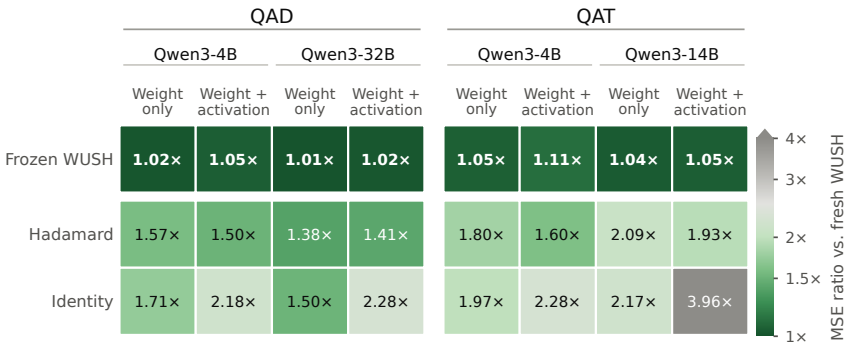

Figure 13: Output MSE of each transform relative to a WUSH transform recomputed on the final weights (median over layers, INT3). The frozen WUSH transform remains within 1.11× of a fresh one, well below Hadamard and Identity.

E THEORETICAL ANALYSIS

E.1 WUSH: OPTIMAL ADAPTIVE TRANSFORMS FOR JOINT QUANTIZATION

Consider a linear layer with weights $W \in \mathbb{R}^{d_{\text{out}} \times d}$ applied to a batch of inputs $X \in \mathbb{R}^{d_{\text{batch}} \times d}$, stacked as rows, with output $XW^\top \in \mathbb{R}^{d_{\text{batch}} \times d_{\text{out}}}$. In joint post-training quantization (PTQ), both operands are quantized and the output is approximated by $q_X(X)q_W(W)^\top$, where q_X and q_W denote the activation and weight quantizers. Due to outliers and anisotropic distributions of weight/input matrices, we introduce a pre-quantization transformation via an invertible matrix $T \in \mathbb{R}^{d \times d}$:

$$\widetilde{W} \coloneqq WT^\top, \qquad \widetilde{X} \coloneqq XT^{-1}, \qquad \widetilde{X}\widetilde{W}^\top = XW^\top, \tag{1}$$

where $T^{-\top} \coloneqq (T^{-1})^\top$. The jointly quantized output is

$$\widehat{Y}(W, T, X) \coloneqq q_X(\widetilde{X})\, q_W(\widetilde{W})^\top = q_X(XT^{-1})\, q_W(WT^\top)^\top. \tag{2}$$

Next we define an objective function which will help us to find the optimal transform T. Denote

$$\Delta_W \coloneqq q_W(\widetilde{W}) - \widetilde{W} = q_W(WT^\top) - WT^\top,$$
$$\Delta_X \coloneqq q_X(\widetilde{X}) - \widetilde{X} = q_X(XT^{-1}) - XT^{-1}.$$

Taking into account that $\widetilde{X}\widetilde{W}^\top = XW^\top$, the output quantization error can be written as

$$\big\|\widehat{Y}(W,T,X) - XW^\top\big\|_{\mathrm{F}}^2 = \big\|(\widetilde{X}+\Delta_X)(\widetilde{W}+\Delta_W)^\top - XW^\top\big\|_{\mathrm{F}}^2 = \big\|\widetilde{X}\Delta_W^\top + \Delta_X\widetilde{W}^\top + \Delta_X\Delta_W^\top\big\|_{\mathrm{F}}^2.$$

To analyze the output quantization error further, we consider its expectation, for which we need the following conditions on the quantizers.

Definition E.1 (Multiplicative quantization noise model)**.** *For a given matrix M, quantization is unbiased (i.e., $\mathbb{E}_q[q(M)] = M$) with quantization error $q(M) - M = \Xi \odot M$, where the entries Ξ_{ij} are independent random variables with common fixed variance. Essentially, the noise is zero-mean and scales with the input signal. For the activation and weight quantizers,*

$$q_A(M) = M + \Xi_A \odot M, \qquad \mathbb{E}_q[(\Xi_A)_{ij}] = 0, \qquad \mathbb{E}_q[(\Xi_A)_{ij}^2] = \sigma_A^2, \qquad A \in \{X, W\}, \tag{3}$$

with Ξ_X and Ξ_W independent.

Here $\odot$ denotes entrywise multiplication, and $\mathbb{E}_q$ averages over the quantization noise with the weights, inputs, and transform held fixed. Then, using the noise model, specifically that Δ_X and Δ_W are zero-mean and independent noises, the expected output quantization error can be expanded as follows:

$$\begin{aligned}
&\mathbb{E}_q\big\|\widehat{Y}(W,T,X) - XW^\top\big\|_{\mathrm{F}}^2 \\
&= \mathbb{E}_q\|\widetilde{X}\Delta_W^\top + \Delta_X\widetilde{W}^\top + \Delta_X\Delta_W^\top\|_{\mathrm{F}}^2 \\
&= \mathbb{E}_q\|\widetilde{X}\Delta_W^\top\|_{\mathrm{F}}^2 + \mathbb{E}_q\|\Delta_X\widetilde{W}^\top\|_{\mathrm{F}}^2 + \mathbb{E}_q\|\Delta_X\Delta_W^\top\|_{\mathrm{F}}^2 \\
&= \mathbb{E}_q\,\mathrm{tr}(\Delta_W^\top\Delta_W\widetilde{X}^\top\widetilde{X}) + \mathbb{E}_q\,\mathrm{tr}(\Delta_X^\top\Delta_X\widetilde{W}^\top\widetilde{W}) + \mathbb{E}_q\,\mathrm{tr}(\Delta_W^\top\Delta_W\Delta_X^\top\Delta_X) \\
&= \mathrm{tr}\big(\mathbb{E}_q[\Delta_W^\top\Delta_W]\,\widetilde{X}^\top\widetilde{X}\big) + \mathrm{tr}\big(\mathbb{E}_q[\Delta_X^\top\Delta_X]\,\widetilde{W}^\top\widetilde{W}\big) + \mathrm{tr}\big(\mathbb{E}_q[\Delta_W^\top\Delta_W]\,\mathbb{E}_q[\Delta_X^\top\Delta_X]\big).
\end{aligned}$$

Next, we use the assumption on the quantizers to simplify the terms involving expectation. Since $\Delta_W = \Xi_W \odot \widetilde{W}$ and $\Delta_X = \Xi_X \odot \widetilde{X}$,

$$\mathbb{E}_q[\Delta_W^\top\Delta_W] = \mathbb{E}_q\big[(\Xi_W \odot \widetilde{W})^\top(\Xi_W \odot \widetilde{W})\big] = \sigma_W^2\,\mathrm{Diag}(\widetilde{W}^\top\widetilde{W}) = \sigma_W^2\,\mathrm{Diag}(TW^\top WT^\top),$$
$$\mathbb{E}_q[\Delta_X^\top\Delta_X] = \mathbb{E}_q\big[(\Xi_X \odot \widetilde{X})^\top(\Xi_X \odot \widetilde{X})\big] = \sigma_X^2\,\mathrm{Diag}(\widetilde{X}^\top\widetilde{X}) = \sigma_X^2\,\mathrm{Diag}(T^{-\top}X^\top XT^{-1}),$$

where Diag keeps the diagonal entries of a square matrix and sets its other entries to zero; for instance, the ith diagonal entry of $\mathrm{Diag}(\widetilde{W}^\top\widetilde{W})$ is $\|\widetilde{W}e_i\|_2^2$. Denote

$$M_W \coloneqq W^\top W, \qquad M_X \coloneqq X^\top X, \qquad c_q \coloneqq \sigma_W^2 + \sigma_X^2 + \sigma_W^2\sigma_X^2, \tag{4}$$

and assume that M_W and M_X are positive definite (say via an additional damping term). Using $\mathrm{tr}(\mathrm{Diag}(A)B) = \mathrm{tr}(A\,\mathrm{Diag}(B)) = \mathrm{tr}(\mathrm{Diag}(A)\,\mathrm{Diag}(B)) = \sum_i A_{ii}B_{ii}$ for any square matrices A and B, we simplify the expected error further:

$$\begin{aligned}\mathbb{E}_q\big\|\widehat{Y}(W,T,X) - XW^\top\big\|_{\mathrm{F}}^2 &= \sigma_W^2\,\mathrm{tr}\big(\mathrm{Diag}(TM_WT^\top)\,T^{-\top}M_XT^{-1}\big)\\ &\quad+\sigma_X^2\,\mathrm{tr}\big(\mathrm{Diag}(T^{-\top}M_XT^{-1})\,TM_WT^\top\big)\\ &\quad+\sigma_W^2\sigma_X^2\,\mathrm{tr}\big(\mathrm{Diag}(TM_WT^\top)\,\mathrm{Diag}(T^{-\top}M_XT^{-1})\big)\\ &= c_q\,\mathrm{tr}\big(\mathrm{Diag}(TM_WT^\top)\,T^{-\top}M_XT^{-1}\big).\end{aligned}$$

Hence the expected output quantization error is

$$\mathbb{E}_q\big\|\widehat{Y}(W,T,X) - XW^\top\big\|_{\mathrm{F}}^2 = c_q\,\phi(W,T,X), \tag{5}$$

and our goal is to choose the transform T such that the following trace is minimized:

$$\phi(W,T,X) \coloneqq \mathrm{tr}\big[\mathrm{Diag}(TM_WT^\top)\,T^{-\top}M_XT^{-1}\big]\,. \tag{6}$$

In the rest of this section W and X are fixed, and we write $\phi(T) \coloneqq \phi(W,T,X)$. The functional depends on (W,X) only through (M_W, M_X), and Lemma E.1 and Theorem E.3 below use only that these are symmetric positive definite matrices of order d. Both results therefore hold verbatim for any such pair in place of (M_W, M_X), and for any order in place of d; this is how they are used in Sections E.5–E.7.

E.2 Lower bound for the quantization error

Lemma E.1 (Lower bound). *Let $\Gamma \coloneqq M_W^{1/2}M_XM_W^{1/2}$. For every invertible T it holds*

$$\phi(T) \geq \frac{1}{d}\big(\mathrm{tr}\,\Gamma^{1/2}\big)^2.$$

Proof. Let us whiten the weights, namely we search the transform T in the form $T = T_1L_W^{-1}$, where $M_W = L_WL_W^\top$ is a square factorization. We take $L_W = M_W^{1/2}$; a Cholesky factor works equally well, since it replaces Γ below by an orthogonally similar matrix. Then we have

$$\begin{aligned}\phi(T) &= \mathrm{tr}\,\big(\mathrm{Diag}(TM_WT^\top)\,T^{-\top}M_XT^{-1}\big)\\ &= \mathrm{tr}\,\big(\mathrm{Diag}(T_1L_W^{-1}M_WL_W^{-\top}T_1^\top)\,T_1^{-\top}L_W^\top M_XL_WT_1^{-1}\big)\\ &= \mathrm{tr}\,\big(\mathrm{Diag}(T_1T_1^\top)\,T_1^{-\top}\Gamma T_1^{-1}\big)\\ &= \mathrm{tr}\,\big(T_1^{-1}\,\mathrm{Diag}(T_1T_1^\top)T_1^{-\top}\,\Gamma\big)\,,\end{aligned}$$

since $L_W^\top M_XL_W = \Gamma$ for this choice of L_W. Denote $D = \mathrm{Diag}(T_1T_1^\top)$ and $Z = T_1^{-1}DT_1^{-\top}$; then the objective becomes $\mathrm{tr}(Z\Gamma)$. Note that

$$\mathrm{tr}(Z^{-1}) = \mathrm{tr}(T_1^\top D^{-1}T_1) = \mathrm{tr}(T_1T_1^\top D^{-1}) = \mathrm{tr}(\mathrm{Diag}(T_1T_1^\top)D^{-1}) = \mathrm{tr}(I) = d.$$

Since the feasible set is relaxed, the minimum can be smaller, implying that

$$\min_{T\ \text{invertible}}\phi(T) \geq \min_{Z\succ 0,\ \mathrm{tr}(Z^{-1})=d}\mathrm{tr}(Z\Gamma).$$

The obtained optimization problem can be solved with Lagrange multipliers. (Replacing the constraint by $\mathrm{tr}(Z^{-1}) \leq d$ gives a convex problem with the same optimal value, because shrinking Z shows that the constraint is active; hence the stationary point below is the minimizer.) Indeed, the optimality condition for the Lagrangian $\mathcal{L}(Z,\lambda) = \mathrm{tr}(Z\Gamma) + \lambda(\mathrm{tr}(Z^{-1}) - d)$ with multiplier λ implies

$$\nabla_Z\mathcal{L} = \Gamma - \lambda Z^{-2} = 0 \implies Z = \sqrt{\lambda}\,\Gamma^{-1/2}.$$

We can compute λ from the equality constraint:

$$d = \mathrm{tr}(Z^{-1}) = \frac{1}{\sqrt{\lambda}}\,\mathrm{tr}(\Gamma^{1/2}) \implies \sqrt{\lambda} = \frac{1}{d}\,\mathrm{tr}(\Gamma^{1/2}).$$

Hence the optimal $Z = \frac{1}{d}\,\mathrm{tr}(\Gamma^{1/2})\,\Gamma^{-1/2}$ gives us the optimal/minimal value $\mathrm{tr}(Z\Gamma) = \frac{1}{d}(\mathrm{tr}(\Gamma^{1/2}))^2$. □

Remark E.2 (A direct argument)**.** *The bound also follows without the relaxation, and it holds for merely positive semidefinite M_W and M_X. Let $M_W = L_W L_W^\top$ and $M_X = L_X L_X^\top$ with square factors, and let u_i and v_i be the columns of $L_W^\top T^\top$ and $L_X^\top T^{-1}$. Then $L_W^\top L_X = \sum_i u_i v_i^\top$ and $\phi(T) = \sum_i \|u_i\|^2 \|v_i\|^2$. The squared singular values of $L_W^\top L_X$ are the eigenvalues of Γ. Writing $\|\cdot\|_*$ for the nuclear norm, the sum of singular values, its triangle inequality and Cauchy–Schwarz give*

$$\operatorname{tr} \Gamma^{1/2} = \|L_W^\top L_X\|_* \le \sum_i \|u_i\| \|v_i\| \le \sqrt{d\,\phi(T)}.$$

E.3 Construction of optimal transforms (WUSH)

Theorem E.3 (WUSH transform)**.** *Let $M_W = L_W L_W^\top$ and $M_X = L_X L_X^\top$ with square invertible factors (e.g., Cholesky), let $L_W^\top L_X = U S V^\top$ be the singular value decomposition, and let H be the normalized Hadamard matrix of order d, assumed to exist (see Remark E.4). Then*

$$\boxed{T^\star \coloneqq H S^{1/2} U^\top L_W^{-1}} \tag{7}$$

balances the two moments, $T^\star M_W T^{\star\top} = T^{\star-\top} M_X T^{\star-1} = H S H^\top$, and attains the lower bound of Lemma E.1:

$$\phi(T^\star) = \frac{1}{d}(\operatorname{tr} S)^2 = \frac{1}{d}\big(\operatorname{tr} \Gamma^{1/2}\big)^2 = \min_{T \text{ invertible}} \phi(T).$$

Proof. We have

$$\begin{aligned}
T^\star M_W T^{\star\top} &= H S^{1/2} U^\top L_W^{-1} \cdot L_W L_W^\top \cdot L_W^{-\top} U S^{1/2} H^\top = H S H^\top, \\
T^{\star-\top} M_X T^{\star-1} &= H S^{-1/2} U^\top L_W^\top \cdot L_X L_X^\top \cdot L_W U S^{-1/2} H^\top \\
&= H S^{-1/2} U^\top (L_W^\top L_X)(L_W^\top L_X)^\top U S^{-1/2} H^\top \\
&= H S^{-1/2} U^\top (U S V^\top)(V S U^\top) U S^{-1/2} H^\top = H S H^\top.
\end{aligned}$$

Hence, the objective becomes

$$\begin{aligned}
\phi(T^\star) &= \operatorname{tr}\left(\operatorname{Diag}(T^\star M_W T^{\star\top})\, \operatorname{Diag}(T^{\star-\top} M_X T^{\star-1})\right) \\
&= \operatorname{tr}\left(\operatorname{Diag}(H S H^\top)^2\right) = \sum_{i=1}^d (H S H^\top)_{ii}^2 = \sum_{i=1}^d \left(\sum_{j=1}^d H_{ij} S_{jj} H_{ij}\right)^2 \\
&= \sum_{i=1}^d \left(\sum_{j=1}^d \frac{1}{d} S_{jj}\right)^2 = \sum_{i=1}^d \frac{1}{d^2} (\operatorname{tr}(S))^2 = \frac{1}{d}(\operatorname{tr}(S))^2 = \frac{1}{d}(\operatorname{tr}(\Gamma^{1/2}))^2,
\end{aligned}$$

where we used $H_{ij}^2 = 1/d$ and, in the last step, that the squared singular values of $L_W^\top L_X$ are the eigenvalues of Γ (Remark E.2). By Lemma E.1, this value is the minimum. □

Remark E.4 (Equalizers)**.** *The proof uses H only through its orthogonality and $\operatorname{Diag}(H S H^\top) = (\operatorname{tr} S/d) I_d$. Hence H can be replaced by any orthogonal P with $\operatorname{Diag}(P S P^\top) = (\operatorname{tr} S/d) I_d$. Such an equalizing matrix exists in every dimension: choose a unit vector with Rayleigh quotient $\operatorname{tr} S/d$, then repeat on its orthogonal complement, whose average eigenvalue is unchanged. Since $L_W^\top L_X V = U S$, the transform can also be computed from the activation-side factor, $T^\star = P S^{-1/2} V^\top L_X^\top$.*

Remark E.5 (Riccati equation)**.** *Notice that the optimal transform satisfies*

$$T^\star M_W T^{\star\top} = T^{\star-\top} M_X T^{\star-1} \iff (T^{\star\top} T^\star) M_W (T^{\star\top} T^\star) = M_X,$$

which is the Riccati equation with respect to $T^{\star\top} T^\star$. Its unique positive definite solution is given by

$$T^{\star\top} T^\star = M_W^{-1} \# M_X \coloneqq M_W^{-1/2} \left(M_W^{1/2} M_X M_W^{1/2}\right)^{1/2} M_W^{-1/2},$$

where the matrix geometric mean is defined via $A \# B = A^{1/2}(A^{-1/2} B A^{-1/2})^{1/2} A^{1/2}$. If A and B commute, $AB = BA$, then $A \# B = A^{1/2} B^{1/2}$.

Remark E.6 (Singular moments and damping)**.** *The positive-definiteness assumption is needed for the displayed invertible minimizer. For singular M_W and $M_X \succ 0$, the infimum formula $\inf_T \phi(T) = \frac{1}{d}(\operatorname{tr}\Gamma^{1/2})^2$ still holds, but it need not be attained. A practical damped construction uses $\overline{M}_W = M_W + \lambda I$, $\lambda > 0$. The functional is nondecreasing in the weight moment in the positive-semidefinite order. Thus at the transform computed from $\overline{M}_W$, its undamped value is bounded by the damped optimum. Letting $\lambda \downarrow 0$, with Remark E.2 for the matching lower bound, proves the infimum formula.*

E.4 QAT FOR LINEAR REGRESSION WITH A QUANTIZED FORWARD PASS

From here on, we switch from PTQ, where the weights are given, to quantization-aware training (QAT), where they are learned through the quantized forward pass (2). The first setting we consider is QAT of a linear regression model. The goal is to learn a model that predicts accurately using low-precision weight–activation multiplications and can be exported with fixed low-precision weights. Training maintains full-precision master weights and uses quantized operands in the forward multiplication.

Data and notation. Let $(x, y) \sim \mathcal{D}$, with $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^{d_{\text{out}}}$. A batch consists of d_{batch} independent samples, stacked as rows: the inputs form $X \in \mathbb{R}^{d_{\text{batch}} \times d}$, as in Section E.1, and the responses form $Y \in \mathbb{R}^{d_{\text{batch}} \times d_{\text{out}}}$. Here $W \in \mathbb{R}^{d_{\text{out}} \times d}$ denotes the master weights, $XW^\top$ is the full-precision prediction, and Y contains the observed responses. We use the regression model

$$y = W_* x + \varepsilon, \quad \mathbb{E}[\varepsilon \mid x] = 0, \quad \Sigma_X \coloneqq \mathbb{E}[X^\top X] = d_{\text{batch}}\mathbb{E}[xx^\top] \succeq 0, \quad \mathbb{E}\|\varepsilon\|^2 = \frac{\sigma^2}{d_{\text{batch}}}, \tag{8}$$

with finite second moments. The matrix W_* describes the data-generating linear map. All output errors use the ordinary squared Euclidean or Frobenius norm.

Transforms and quantizers. The invertible transform T, the transformed operands (1), and the quantizers q_X and q_W are those of Section E.1. Their formats, grouping rules, and scale-selection rules are fixed parts of the setup. In formulas, a quantized tensor denotes its numerical values after decoding; its stored representation consists of low-precision codes and the required scales. The quantized forward prediction is $\widehat{Y}(W, T, X)$ from (2), now evaluated at the master weights.

Optimization objective and exported model. The objective we aim to minimize measures the prediction error of the jointly quantized forward pass:

$$\boxed{\inf_{W,T} J(W, T) \coloneqq \mathbb{E}_{X,Y,q}\Big[\big\|q_X(XT^{-1})q_W(WT^\top)^\top - Y\big\|_{\text{F}}^2\Big].} \tag{9}$$

The expectation is over fresh mini-batch data (X, Y) and any quantization randomness q.

What is retained after training. After K training updates, freeze a final transform T_K and export

$$\widehat{W}_K \coloneqq q_W(W_K T_K^\top). \tag{10}$$

For each new input batch X, we compute

$$\widehat{Y}_K(X) = q_X(XT_K^{-1})\widehat{W}_K^\top. \tag{11}$$

The master weights W_K are not needed for inference. Weight transformation and quantization are performed once at export; activation transformation and quantization are performed on incoming data. The central multiplication uses low-precision operands.

E.5 RISK DECOMPOSITION AND THE ROLE OF WUSH

To analyze the objective (9) with joint quantization, we adopt the idealized multiplicative-noise model (3) of Definition E.1 for both quantizers. Here $\mathbb{E}_q$ fixes the data (X, Y), weights, and transform. Conditional on these quantities, noise entries are independent within each operand and across the two quantizers. The relative noise variances σ_X^2 and σ_W^2 are fixed and distinct from the observation-noise variance σ^2 in (8).

Prediction error and reconstruction error. Define

$$R(W, T) \coloneqq \mathbb{E}_{X,q}\|\widehat{Y}(W, T, X) - XW^\top\|_{\text{F}}^2. \tag{12}$$

Under (3), $\mathbb{E}_q[\widehat{Y}(W,T,X)] = XW^\top$. Expanding the squared prediction error therefore gives the exact identity

$$\boxed{J(W,T) = L(W) + R(W,T),} \tag{13}$$

where the full-precision loss is

$$L(W) \coloneqq \mathbb{E}_{X,Y}\|XW^\top - Y\|_{\mathrm{F}}^2 = \|(W - W_*)\Sigma_X^{1/2}\|_{\mathrm{F}}^2 + \sigma^2. \tag{14}$$

The cross term vanishes after averaging over quantization conditional on the data, weights, and transform. The term R is part of the quantized prediction objective itself.

WUSH as the transform subproblem. For a realized batch, take M_W, M_X, and c_q as in (4). Thus $M_X = X^\top X$ is random and $\Sigma_X = \mathbb{E}[M_X]$ is its population counterpart. Conditional on the batch, the computation of Section E.1 gives $\mathbb{E}_q\|\widehat{Y}(W,T,X) - XW^\top\|_{\mathrm{F}}^2 = c_q\,\phi(W,T,X)$, see (5), with the batch WUSH functional (6). Since $\phi(W,T,X)$ is linear in M_X, taking expectation over the batch gives

$$R(W,T) = c_q\,\phi(W,T), \qquad \phi(W,T) \coloneqq \mathrm{tr}\big[\mathrm{Diag}(TM_WT^\top)\,T^{-\top}\Sigma_XT^{-1}\big]. \tag{15}$$

At fixed W, $L(W)$ is independent of T. Consequently, under $c_q > 0$, minimizing $J(W,T)$ over invertible $T \in \mathbb{R}^{d\times d}$ is exactly the WUSH reconstruction subproblem of minimizing $\phi(W,T)$, that is, the problem of Section E.1 with M_X replaced by Σ_X; for $M_W, \Sigma_X \succ 0$ it is solved by the transform of Theorem E.3. If $c_q = 0$, every admissible transform gives the same risk.

This identifies WUSH's role in QAT: it selects a transform that reduces the quantization contribution to the prediction loss at the current master weights. On the other hand, weight training must account for the full objective J, since both L and R depend on W.

E.6 Fixed-transform convergence of SGD–STE

We now analyze the training pipeline of Section E.4 with an invertible transform T held fixed: quantized forward multiplication, full-precision backward arithmetic, and straight-through estimation (STE) through weight quantization. At step t, draw a fresh batch (X_t, Y_t) and set

$$\begin{aligned} G_t &\coloneqq 2\big(\widehat{Y}(W_t,T,X_t) - Y_t\big)^\top q_X(X_tT^{-1})T, \\ W_{t+1} &= W_t - \eta_t G_t. \end{aligned} \tag{16}$$

The same quantized activation is used in the forward pass and in G_t; T and the quantizer scales are held fixed in backward. Let $\mathbb{E}_t$ be the expectation conditional on all randomness before the current draws.

The mean update and the two reference weights. Under the quantization model (3), define

$$\begin{aligned} P_T &\coloneqq T^\top\,\mathrm{Diag}(T^{-\top}\Sigma_XT^{-1})T, \\ H_T &\coloneqq \Sigma_X + c_qP_T, \qquad H_{\mathrm{s}} \coloneqq \Sigma_X + \sigma_X^2P_T. \end{aligned} \tag{17}$$

These are $d \times d$ matrices. Since $\phi(W,T) = \mathrm{tr}(WP_TW^\top)$, the expected loss $J = L + c_q\phi$ is quadratic in W. Its gradient and the mean STE update are

$$\nabla_W J(W,T) = 2(WH_T - W_*\Sigma_X), \qquad \mathbb{E}_tG_t = 2(W_tH_{\mathrm{s}} - W_*\Sigma_X). \tag{18}$$

The first identity is the gradient of the expected loss, whereas the second describes the conditional mean of the actual update (16). The discrepancy is precisely the missing term $2(c_q - \sigma_X^2)W_tP_T$ in the mean STE gradient. This changes the implicit objective targeted by the STE update and, consequently, its minimizer.

Assume $\Sigma_X \succ 0$ throughout this section. The minimizer of $J(\cdot,T)$ and the zero of the mean STE update are, respectively,

$$W_T \coloneqq W_*\Sigma_XH_T^{-1}, \qquad W_{\mathrm{s}} \coloneqq W_*\Sigma_XH_{\mathrm{s}}^{-1}. \tag{19}$$

The first minimizes $J = L + c_q\phi$, whereas the second minimizes $L + \sigma_X^2\phi$. Both depend on the fixed transform. In the absence of quantization noise, i.e., when $\sigma_X^2 = \sigma_W^2 = 0$, both reference weights

reduce to the data-generating matrix W_*, which minimizes the full-precision loss L. For a positive definite matrix N, define $\|U\|_N^2 := \|UN^{1/2}\|_{\mathrm{F}}^2$.

Separating the persistent errors. To display both sources of residual loss, compare with the full-precision minimum $L(W_*) = \sigma^2$. Define

$$\begin{aligned} \Delta_{\mathrm{FQ}}(T) &:= J(W_T, T) - L(W_*) = L(W_T) - L(W_*) + R(W_T, T), \\ \Delta_{\mathrm{STE}}(T) &:= J(W_{\mathrm{s}}, T) - J(W_T, T) = \|W_{\mathrm{s}} - W_T\|_{H_T}^2. \end{aligned} \tag{20}$$

The first is the cost of forward quantization after optimizing the master weights; it includes the shift from W_* to W_T. The second is the additional cost of the STE mean update targeting W_{s}. Both are nonnegative.

Assumption E.7 (Variance growth for STE)**.** *For a fixed invertible transform T, there exist finite constants $A_T, B_T \geq 0$, independent of the current master weights, such that the centered STE noise satisfies*

$$\mathbb{E}_t\|G_t - \mathbb{E}_t G_t\|_{H_{\mathrm{s}}}^2 \leq A_T\|W_t - W_{\mathrm{s}}\|_{H_{\mathrm{s}}}^2 + B_T. \tag{21}$$

Here A_T controls how the STE variance may grow with the distance from the mean-update fixed point W_{s}, while B_T bounds the variance at W_{s}. Sufficient moment conditions for this assumption, together with the resulting step-size constants, are given in the proof.

Theorem E.8 (Fixed-T SGD–STE: rate and residuals)**.** *Under the setup above and Assumption E.7, let W_0 be deterministic. There are constants $\eta_{\max} > 0$ and $C_T, V_T \geq 0$, specified in the proof and independent of K and η, such that every constant step $0 < \eta \leq \eta_{\max}$ satisfies*

$$\boxed{\mathbb{E}J(W_K, T) - L(W_*) \leq C_T\big(1 - 2\eta\lambda_{\min}(H_{\mathrm{s}})\big)^K + \eta V_T + \Delta_{\mathrm{FQ}}(T) + \Delta_{\mathrm{STE}}(T).} \tag{22}$$

For $\eta_t = 1/[\lambda_{\min}(H_{\mathrm{s}})(t + t_0)]$ and $t_0 \geq 1/[\lambda_{\min}(H_{\mathrm{s}})\eta_{\max}]$, the iterates converge to W_{s} in mean square and

$$\mathbb{E}J(W_K, T) - L(W_*) = \Delta_{\mathrm{FQ}}(T) + \Delta_{\mathrm{STE}}(T) + O(1/K). \tag{23}$$

The two residuals obey the common reconstruction bound

$$\boxed{0 \leq \Delta_{\mathrm{FQ}}(T) + \Delta_{\mathrm{STE}}(T) \leq R(W_*, T) = c_q\phi(W_*, T).} \tag{24}$$

Proof. Mean dynamics. Fix W, X, Y, and let G be the estimator in (16) for these values. Average over weight noise, then over activation noise. Using $\mathbb{E}_q[q_X(\widetilde{X})^\top q_X(\widetilde{X})] = \widetilde{X}^\top\widetilde{X} + \sigma_X^2\,\mathrm{Diag}(\widetilde{X}^\top\widetilde{X})$ gives

$$\mathbb{E}_q G = 2(WX^\top X - Y^\top X) + 2\sigma_X^2 W T^\top\,\mathrm{Diag}(T^{-\top}X^\top X T^{-1})T.$$

Averaging over the batch proves (18), since $\mathbb{E}[Y^\top X] = W_*\Sigma_X$. Completing the square in J proves $J(W, T) - J(W_T, T) = \|W - W_T\|_{H_T}^2$ and the residual identities.

Stochastic recursion and constants. For the proof, put

$$m = 2\lambda_{\min}(H_{\mathrm{s}}), \qquad F_t = \|W_t - W_{\mathrm{s}}\|_{H_{\mathrm{s}}}^2, \qquad r = \lambda_{\max}(H_{\mathrm{s}}^{-1/2} H_T H_{\mathrm{s}}^{-1/2}).$$

Thus $H_T \preceq r H_{\mathrm{s}}$. The constants in the statement can be taken as

$$\begin{aligned} \eta_{\max} &= \min\left\{\frac{1}{2\lambda_{\max}(H_{\mathrm{s}})}, \frac{m}{m^2 + A_T}\right\}, \\ C_T &= rF_0 + 2\sqrt{r\Delta_{\mathrm{STE}}(T)F_0}, \qquad V_T = \frac{rB_T}{m}. \end{aligned} \tag{25}$$

Let $\xi_t = G_t - \mathbb{E}_t G_t$. The mean identity yields

$$W_{t+1} - W_{\mathrm{s}} = (W_t - W_{\mathrm{s}})(I_d - 2\eta_t H_{\mathrm{s}}) - \eta_t\xi_t, \qquad \mathbb{E}_t\xi_t = 0.$$

The cross term in the squared H_{s}-norm vanishes conditionally. For $\eta_t \leq \eta_{\max}$, the eigenvalues of $I_d - 2\eta_t H_{\mathrm{s}}$ lie in $[0, 1 - m\eta_t]$. Hence

$$\mathbb{E}_t F_{t+1} \leq \big[(1 - m\eta_t)^2 + A_T\eta_t^2\big]F_t + B_T\eta_t^2 \leq (1 - m\eta_t)F_t + B_T\eta_t^2. \tag{26}$$

For a constant step, unrolling gives

$$\mathbb{E}F_K \le (1-m\eta)^K F_0 + \frac{\eta B_T}{m}\big[1-(1-m\eta)^K\big].$$

For $\eta_t = 2/[m(t+t_0)]$, induction in (26) gives

$$\mathbb{E}F_K \le \frac{\max\{t_0 F_0, 4B_T/m^2\}}{K+t_0}. \tag{27}$$

Indeed, with $s = t+t_0 \ge 2$, a bound $\mathbb{E}F_t \le C/s$ propagates to $\mathbb{E}F_{t+1} \le C(s-1)/s^2 \le C/(s+1)$ whenever $C \ge 4B_T/m^2$.

Conversion to prediction loss. For deterministic steps, the mean error satisfies

$$\mathbb{E}[W_K - W_{\mathrm{s}}] = (W_0 - W_{\mathrm{s}}) \prod_{t=0}^{K-1} (I_d - 2\eta_t H_{\mathrm{s}}).$$

Expanding the squared H_T-distance around W_{s} gives

$$\begin{aligned}\mathbb{E}J(W_K,T) - L(W_*) = \Delta_{\mathrm{FQ}}(T) + \Delta_{\mathrm{STE}}(T) + \mathbb{E}\|W_K - W_{\mathrm{s}}\|_{H_T}^2 \\ + 2\langle \mathbb{E}[W_K - W_{\mathrm{s}}]H_T, W_{\mathrm{s}} - W_T\rangle_{\mathrm{F}}.\end{aligned}$$

Using $H_T \preceq rH_{\mathrm{s}}$ and Cauchy–Schwarz, the absolute difference from the sum of residuals is at most

$$r\mathbb{E}F_K + 2\sqrt{r\Delta_{\mathrm{STE}}(T)F_0}\prod_{t=0}^{K-1}(1-m\eta_t). \tag{28}$$

The constant-step bound follows. For decreasing steps, $1-2/s \le (s/(s+1))^2$ gives a product at most $(t_0/(K+t_0))^2$. Combining this with (27) proves (23); positive definiteness of H_{s} also gives mean-square convergence of the weights.

Reconstruction control of the residuals. Since W_{s} minimizes $L + \sigma_X^2\phi$,

$$L(W_{\mathrm{s}}) - L(W_*) + \sigma_X^2\phi(W_{\mathrm{s}},T) \le \sigma_X^2\phi(W_*,T).$$

For $\sigma_X^2 > 0$, nonnegativity of $L(W_{\mathrm{s}}) - L(W_*)$ implies $\phi(W_{\mathrm{s}},T) \le \phi(W_*,T)$. For $\sigma_X^2 = 0$, this also holds because $W_{\mathrm{s}} = W_*$. Adding $(c_q - \sigma_X^2)\phi(W_{\mathrm{s}},T)$ to the preceding inequality therefore proves (24). More specifically, W_T minimizes J and $(W_{\mathrm{s}} - W_T)H_T = (c_q - \sigma_X^2)W_{\mathrm{s}}P_T$. Thus, for $c_q > 0$,

$$\begin{aligned}\Delta_{\mathrm{FQ}}(T) &\le c_q\phi(W_*,T), \\ \Delta_{\mathrm{STE}}(T) &= (c_q-\sigma_X^2)^2\|W_{\mathrm{s}}P_T H_T^{-1/2}\|_{\mathrm{F}}^2 \\ &\le \frac{(c_q-\sigma_X^2)^2}{c_q}\phi(W_{\mathrm{s}},T) \le \frac{(c_q-\sigma_X^2)^2}{c_q}\phi(W_*,T).\end{aligned} \tag{29}$$

The matrix inequality $P_T H_T^{-1} P_T \preceq P_T/c_q$ follows from $H_T \succeq c_q P_T$. If $c_q = 0$, both residuals vanish.

Sufficient moments. Condition (21) holds, for example, when the quantization noises have fixed distributions independent of the data and W, the input and activation relative noise have finite fourth moments, and $\mathbb{E}[\|\varepsilon\|^2\|x\|^2] < \infty$. For fixed draws the STE gradient is affine in W, with a linear coefficient having finite second moment; its value at W_{s} also has finite second moment. Splitting at W_{s} and applying the squared triangle inequality gives finite A_T, B_T. The weight-noise second moments already assumed in (3) suffice here. □

What the rate says. With constant steps, the bound consists of a geometric transient, a step-size-dependent sampling term ηV_T, and two persistent residuals. Decreasing steps remove the sampling term and leave exactly $\Delta_{\mathrm{FQ}}(T) + \Delta_{\mathrm{STE}}(T)$ above $L(W_*)$. Relative to the quantized optimum $J(W_T,T)$, the persistent residual is exactly $\Delta_{\mathrm{STE}}(T)$.

Which part WUSH optimizes. Both residuals depend on T through P_T and the two reference weights. Equations (24) and (29) give a precise common control: their sum, and the separate bounds,

are multiples of $\phi(W_*, T)$. Optimizing these displayed residual bounds over T is therefore exactly the WUSH reconstruction subproblem with weight moment $M_W = W_*^\top W_*$ and activation moment Σ_X, solved by Theorem E.3 for this pair (with the damping of Remark E.6 if $W_*^\top W_*$ is singular). This is optimality of the residual *bounds*; it does not require the two exact residuals to have the same minimizing transform.

In case of weight-only quantization, namely $\sigma_X^2 = 0$, we have $W_\mathrm{s} = W_*$ and

$$\Delta_{\mathrm{FQ}}(T) + \Delta_{\mathrm{STE}}(T) = R(W_*, T).$$

In this case WUSH optimizes the exact asymptotic loss floor. When $\sigma_W^2 = 0$, the STE residual vanishes because $H_\mathrm{s} = H_T$. With both operands quantized, the common bound above applies.

The transient and sampling constants also depend on T, through the curvature H_s and the variance of the actual STE gradient. Reconstruction optimality alone does not minimize those constants. Finally, the reference W_* in the residual bound is the regression target, and T remains fixed throughout this theorem. Transforms computed from the evolving training weights are treated in Section E.7, for weight-only quantization of a nonlinear model.

E.7 Nonlinear QAT with WUSH weight transforms

Theorem E.8 gives an exact description of the mean STE dynamics for linear regression. This section extends the analysis to a smooth nonlinear model, with quantization of the weights only. The activations and backward arithmetic remain in full precision. The central quantity is a curvature-weighted distortion D_t: it controls both the change in the gradient during training and the change in loss when the quantized model is exported.

We first state a convergence result in terms of D_t, without specifying the transform. We then use the structure of a linear layer inside a nonlinear model to bound D_t by a sum of WUSH functionals. Under the multiplicative-noise model, WUSH minimizes this bound at the current weights.

E.8 Weight-only QAT and convergence

Let $\theta \in \mathbb{R}^p$ collect the master weight parameters and let $f(\theta)$ be the full-precision population loss, with $f^\star \coloneqq \inf_\theta f(\theta) > -\infty$. For an invertible weight transform $\mathcal{T}$, define

$$Q_{\mathcal{T}}(\theta) \coloneqq \mathcal{T}^{-1} q_W(\mathcal{T}\theta), \qquad J(\theta, \mathcal{T}) \coloneqq \mathbb{E}_q f(Q_{\mathcal{T}}(\theta)). \tag{30}$$

Here J denotes the expected loss of the nonlinear quantized model, extending the risk notation of (9) to this weight-only setting. The expectation in f is over fresh evaluation data; the expectation in J additionally averages weight quantization. It is not a reconstruction surrogate.

Starting from deterministic θ_0, the QAT iteration is

$$\widehat{\theta}_t = Q_{\mathcal{T}_t}(\theta_t), \qquad G_t = g(\widehat{\theta}_t; \zeta_t), \qquad \theta_{t+1} = \theta_t - \eta G_t. \tag{31}$$

The transform and quantizer scales are held fixed in backward, and rounding has the identity surrogate derivative. Thus the surrogate Jacobian is $\mathcal{T}_t^{-1} I \mathcal{T}_t = I$: the gradient at the quantized weights is applied to the master weights. This update is generally not the gradient of J. At export, freeze $\mathcal{T}_K$, quantize the final weights once, and retain their codes, scales, and the required transforms. The expectation of this exported model's loss is $\mathbb{E}J(\theta_K, \mathcal{T}_K)$.

Let $\mathcal{F}_t$ contain the history and all information used to choose $\mathcal{T}_t$, including θ_t, and set $\mathbb{E}_t = \mathbb{E}[\cdot \mid \mathcal{F}_t]$. The transform is $\mathcal{F}_t$-measurable; its quantization noise is fresh, and the gradient sample ζ_t is independent of both. A fixed offline transform satisfies this requirement. If transforms are refreshed, statistics from earlier batches or an independent calibration sample preserve it. Reusing the gradient batch to choose $\mathcal{T}_t$ is not covered by this conditional-independence setup.

Assumption E.9 (Matrix smoothness and gradient variance)**.** *There is a fixed symmetric positive definite matrix $\mathcal{H} \in \mathbb{R}^{p\times p}$ such that*

$$\|\nabla f(u) - \nabla f(v)\|_{\mathcal{H}^{-1}} \le \|u - v\|_{\mathcal{H}} \qquad (u, v \in \mathbb{R}^p), \tag{32}$$

where $\|v\|_{\mathcal{H}}^2 = v^\top \mathcal{H} v$. The stochastic gradient oracle satisfies

$$\mathbb{E}_\zeta g(v; \zeta) = \nabla f(v), \qquad \mathbb{E}_\zeta \|g(v; \zeta) - \nabla f(v)\|^2 \le \sigma_g^2. \tag{33}$$

Write $\beta = \lambda_{\max}(\mathcal{H})$. For twice continuously differentiable f, a sufficient condition for (32) is $-\mathcal{H} \preceq \nabla^2 f(v) \preceq \mathcal{H}$ for every v. The scalar σ_g^2 is the sampling variance and is distinct from the quantization variances and regression observation noise.

Define the weight error and its curvature-weighted distortion by

$$e_t \coloneqq \widehat{\theta}_t - \theta_t, \qquad D_t \coloneqq \mathbb{E}_t \|e_t\|_{\mathcal{H}}^2, \qquad \overline{D}_K \coloneqq \frac{1}{K} \sum_{t=0}^{K-1} \mathbb{E} D_t. \tag{34}$$

Under the weight-noise model (3), $\mathbb{E}_t e_t = 0$. This property controls the loss at export; matrix smoothness controls the gradient perturbation during training through D_t.

Theorem E.10 (Nonlinear SGD–STE: convergence and quantized risk)**.** *Assume the setup above and Assumption E.9, with $\mathbb{E}_t e_t = 0$ and finite expectations in the bounds below. Let $0 < \eta \le 1/\beta$ and $F_0 = f(\theta_0) - f^\star$. For every $K \ge 1$,*

$$\boxed{\frac{1}{K} \sum_{t=0}^{K-1} \mathbb{E} \|\nabla f(\theta_t)\|^2 \le \frac{2F_0}{\eta K} + \eta\beta\sigma_g^2 + \beta \overline{D}_K.} \tag{35}$$

If, in addition, f satisfies the Polyak–Łojasiewicz (PL) inequality $\|\nabla f(v)\|^2 \ge 2\mu[f(v) - f^\star]$ with $0 < \mu \le \beta$, put $D_{\max} = \max_{0 \le t \le K} \mathbb{E} D_t$, including the export distortion at $t = K$. Then the exported model satisfies

$$\boxed{\mathbb{E} J(\theta_K, \mathcal{T}_K) - f^\star \le (1 - \eta\mu)^K F_0 + \frac{\eta\beta\sigma_g^2}{2\mu} + \frac{\beta + \mu}{2\mu} D_{\max}.} \tag{36}$$

Each bound contains one quantization contribution, whose coefficient is independent of η. In (36), this term covers both training and export. Decreasing the step size reduces the sampling term but does not by itself reduce the distortion budget. Without PL, the export inequality $\mathbb{E} J(\theta_K, \mathcal{T}_K) \le \mathbb{E} f(\theta_K) + \frac{1}{2}\mathbb{E} D_K$ still holds, but stationarity alone does not imply a global loss guarantee.

Proof. Smoothness and gradient perturbation. Integrating (32) along a line segment gives

$$\begin{aligned} |f(v) - f(u) - \langle \nabla f(u), v - u \rangle| &\le \tfrac{1}{2}\|v - u\|_{\mathcal{H}}^2, \\ \|\nabla f(v) - \nabla f(u)\|^2 &\le \beta \|v - u\|_{\mathcal{H}}^2. \end{aligned} \tag{37}$$

For the stated Hessian sufficient condition, the average Hessian along that segment has eigenvalues in $[-1, 1]$ after congruence by $\mathcal{H}^{-1/2}$; its squared action therefore proves (32).

Set $h_t = \nabla f(\theta_t)$ and $\widehat{h}_t = \nabla f(\widehat{\theta}_t)$. By (37), $\mathbb{E}_t \|\widehat{h}_t - h_t\|^2 \le \beta D_t$. The conditional oracle assumptions give $\mathbb{E}_t G_t = \mathbb{E}_t \widehat{h}_t$ and $\mathbb{E}_t \|G_t\|^2 \le \mathbb{E}_t \|\widehat{h}_t\|^2 + \sigma_g^2$.

Descent and stationarity. The scalar smoothness bound and the identity $2\langle h, \widehat{h} \rangle = \|h\|^2 + \|\widehat{h}\|^2 - \|\widehat{h} - h\|^2$ yield

$$\begin{aligned} \mathbb{E}_t f(\theta_{t+1}) &\le f(\theta_t) - \eta \mathbb{E}_t \langle h_t, \widehat{h}_t \rangle + \tfrac{1}{2}\eta^2\beta(\mathbb{E}_t \|\widehat{h}_t\|^2 + \sigma_g^2) \\ &= f(\theta_t) - \tfrac{\eta}{2}\|h_t\|^2 + \tfrac{\eta}{2}\mathbb{E}_t \|\widehat{h}_t - h_t\|^2 - \tfrac{\eta}{2}(1 - \eta\beta)\mathbb{E}_t \|\widehat{h}_t\|^2 + \tfrac{1}{2}\eta^2\beta\sigma_g^2. \end{aligned}$$

Since $\eta\beta \le 1$, dropping the nonpositive term and bounding the gradient perturbation gives

$$\mathbb{E}_t f(\theta_{t+1}) \le f(\theta_t) - \tfrac{\eta}{2}\|h_t\|^2 + \tfrac{1}{2}\eta\beta D_t + \tfrac{1}{2}\eta^2\beta\sigma_g^2. \tag{38}$$

Taking expectations, summing, and using $f(\theta_K) \ge f^\star$ proves (35).

Quantized risk. Applying (37) to the independent export quantization and using its zero conditional mean gives

$$|J(\theta_K, \mathcal{T}_K) - f(\theta_K)| \le \tfrac{1}{2} D_K. \tag{39}$$

Under PL, (38) implies a recursion for the expected full-precision loss gap with contraction $1 - \eta\mu$ and forcing $\frac{1}{2}\eta\beta D_{\max} + \frac{1}{2}\eta^2\beta\sigma_g^2$. Unrolling the recursion and using $\sum_{j=0}^{K-1}(1 - \eta\mu)^j \le 1/(\eta\mu)$ gives

$$\mathbb{E} f(\theta_K) - f^\star \le (1 - \eta\mu)^K F_0 + \frac{\eta\beta\sigma_g^2}{2\mu} + \frac{\beta}{2\mu} D_{\max}.$$

Adding (39) and using $\mathbb{E} D_K \le D_{\max}$ proves (36). □

E.9 Layer structure and the WUSH optimum

We retain the convention of Section E.1: $W \in \mathbb{R}^{d_{\text{out}} \times d}$ and a batch layer output $XW^\top$. Partition the input coordinates of each layer into blocks B of size g_B, with B ranging over all layers. Write $W_B = W_{:,B}$, $X_B = X_{:,B}$, and $w_{B,j} = W_{j,B}^\top \in \mathbb{R}^{g_B}$ for output row j. The parameter ordering for each layer is $\mathrm{vec}(W^\top)$, where vec stacks matrix columns. The same $T_B \in \mathbb{R}^{g_B \times g_B}$ acts on every row of W_B; the global $\mathcal{T}$ is block diagonal with these repeated blocks, in the corresponding ordering of θ. In matrix form,

$$\widehat{W}_B = q_W(W_B T_B^\top) T_B^{-\top}, \qquad X_B \widehat{W}_B^\top = (X_B T_B^{-1}) q_W(W_B T_B^\top)^\top. \tag{40}$$

Only the weight operand is quantized. The inverse transform is applied to activations in full precision.

Let $\mathcal{H}_{B,j}$ be the principal block of $\mathcal{H}$ on the coordinates of $w_{B,j}$. The structural requirement is

$$\mathcal{H}_{B,j} \preceq a_{B,j} C_B, \qquad a_{B,j} > 0, \quad C_B \succ 0. \tag{41}$$

The constants $a_{B,j}$ measure output sensitivity, while C_B describes the geometry of the layer inputs. Define the weighted weight moment and its WUSH functional by

$$\begin{aligned} M_{B,t} &\coloneqq \sum_j a_{B,j} w_{B,j,t} w_{B,j,t}^\top, \\ \phi_{B,t}(T) &\coloneqq \mathrm{tr}\left[\mathrm{Diag}(T M_{B,t} T^\top) T^{-\top} C_B T^{-1}\right]. \end{aligned} \tag{42}$$

This is the WUSH functional (6) in dimension g_B, with the pair $(M_{B,t}, C_B)$ in place of (M_W, M_X). No block-diagonal assumption on the whole matrix $\mathcal{H}$ is imposed.

A nonlinear model giving this structure. Consider a trainable linear layer followed by a fixed nonlinear loss or network, so that $f(W) = \mathbb{E}_\zeta h_\zeta(X_\zeta W^\top)$, with inputs X_ζ independent of W. Suppose there is $A \succ 0$ such that

$$|\langle V, \nabla^2 h_\zeta(Z)[V] \rangle_{\mathrm{F}}| \le \mathrm{tr}(V A V^\top) \qquad \text{for all } Z, V, \zeta. \tag{43}$$

Then $\mathcal{H} = A \otimes \Sigma_X$ is a valid smoothness matrix, provided $\Sigma_X = \mathbb{E}[X_\zeta^\top X_\zeta] \succ 0$ and differentiation under the expectation is valid. Condition (41) holds with $a_{B,j} = A_{jj}$ and $C_B = (\Sigma_X)_{BB}$. This includes nonlinear output losses and a fixed nonlinear tail, not just squared-error regression. For a network whose layers are all trained jointly, (41) must instead be verified for a valid full-network smoothness matrix. An activation moment or a factorized curvature approximation alone does not establish that matrix bound. Calibration estimates can serve as C_B when their discrepancy is covered by the factors $a_{B,j}$.

Proposition E.11 (WUSH minimizes the multiplicative distortion bound)**.** *Assume (41) and the independent weight-noise model (3). Then*

$$D_t \le \mathcal{U}_t(\mathcal{T}_t) \coloneqq \sigma_W^2 \sum_B \phi_{B,t}(T_{B,t}). \tag{44}$$

Equality holds if every principal block in (41) equals $a_{B,j} C_B$.

For the exact WUSH construction, suppose $M_{B,t} \succ 0$ for every block and step under consideration. The WUSH transform of each block is the construction of Theorem E.3 in dimension g_B, applied to $(M_{B,t}, C_B)$ in place of (M_W, M_X): with square invertible factors $M_{B,t} = L_{W,B} L_{W,B}^\top$ and $C_B = L_{C,B} L_{C,B}^\top$,

$$\boxed{T_{B,t}^\star = H_B S_B^{1/2} U_B^\top L_{W,B}^{-1},} \qquad L_{W,B}^\top L_{C,B} = U_B S_B V_B^\top, \tag{45}$$

where $S_B \succ 0$ is diagonal and H_B is a normalized Hadamard matrix of order g_B, if available. Equivalently, $T_{B,t}^\star = H_B S_B^{-1/2} V_B^\top L_{C,B}^\top$, the activation-side form of Remark E.4. The factors depend on t; the time subscript is suppressed here. More generally, H_B can be replaced by any orthogonal equalizer P_B as in Remark E.4; one exists for every block size. The attained minima are

$$\inf_{T \text{ invertible}} \phi_{B,t}(T) = \frac{(\mathrm{tr}\, S_B)^2}{g_B}, \qquad \mathcal{U}_t^\star \coloneqq \inf_{\mathcal{T}} \mathcal{U}_t(\mathcal{T}) = \sigma_W^2 \sum_B \frac{(\mathrm{tr}\, S_B)^2}{g_B}. \tag{46}$$

Write $\mathcal{T}_t^\star$ for the resulting global transform. Under the hypotheses of Theorem E.10, choosing these transforms in (31) gives the explicit bound

$$\frac{1}{K}\sum_{t=0}^{K-1}\mathbb{E}\|\nabla f(\theta_t)\|^2 \le \frac{2F_0}{\eta K} + \eta\beta\sigma_g^2 + \frac{\beta}{K}\sum_{t=0}^{K-1}\mathbb{E}\mathcal{U}_t^\star. \tag{47}$$

Under PL, if the same WUSH rule is used at export, then

$$\mathbb{E}J(\theta_K, \mathcal{T}_K^\star) - f^\star \le (1-\eta\mu)^K F_0 + \frac{\eta\beta\sigma_g^2}{2\mu} + \frac{\beta+\mu}{2\mu}\max_{0\le t\le K}\mathbb{E}\mathcal{U}_t^\star. \tag{48}$$

Thus WUSH minimizes the layerwise distortion bound entering the single quantization term in both convergence guarantees, at fixed weights and curvature statistics.

Proof. From weight noise to the layer functional. Condition on $\mathcal{F}_t$ and let $\xi := q_W(\mathcal{T}_t\theta_t) - \mathcal{T}_t\theta_t$, so that $e_t = \mathcal{T}_t^{-1}\xi$. In a segment (B, j), write $z = T_{B,t}w_{B,j,t}$ and $e_{B,j} = T_{B,t}^{-1}\xi_{B,j}$, where $\xi_{B,j}$ is the corresponding segment of ξ. The entries of ξ are independent and centered, and those of $\xi_{B,j}$ have variances $\sigma_W^2 z_i^2$. Thus different segments have zero cross covariance, even though $\mathcal{H}$ may couple them, and

$$\begin{aligned} D_t &= \sigma_W^2 \sum_{B,j,i} (T_{B,t}^{-\top}\mathcal{H}_{B,j}T_{B,t}^{-1})_{ii}(T_{B,t}w_{B,j,t})_i^2 \\ &\le \sigma_W^2 \sum_B \operatorname{tr}\Big[\operatorname{Diag}(T_{B,t}M_{B,t}T_{B,t}^\top)T_{B,t}^{-\top}C_B T_{B,t}^{-1}\Big]. \end{aligned}$$

This proves (44) and its equality claim.

WUSH optimality. Fix a block. The functional $\phi_{B,t}$ is the WUSH functional of Section E.1 in dimension g_B for the pair $(M_{B,t}, C_B)$, and for this pair $\operatorname{tr}\Gamma^{1/2} = \operatorname{tr} S_B$ (Remark E.2). Hence Lemma E.1 gives $\phi_{B,t}(T) \ge (\operatorname{tr} S_B)^2/g_B$ for every invertible T, and Theorem E.3, with Remark E.4 for a general equalizer, shows that $T_{B,t}^\star$ attains this value. Summing the block minima proves (46); the rate bounds follow from Theorem E.10 and the nonnegativity of its coefficients.

The nonlinear layer example. For a perturbation U of W, the second directional derivative of f is bounded in absolute value, by (43), by

$$\mathbb{E}\operatorname{tr}[(X_\zeta U^\top)A(X_\zeta U^\top)^\top] = \operatorname{tr}(AU\Sigma_X U^\top) = \|\operatorname{vec}(U^\top)\|_{A\otimes\Sigma_X}^2.$$

This proves the claimed two-sided Hessian bound. Its principal block on input coordinates B of output row j is $A_{jj}(\Sigma_X)_{BB}$, as required. □

The positive-definiteness assumption is needed only for the displayed invertible minimizer. By Remark E.6, for singular $M_{B,t}$ the infimum formula (46) still holds, with $\operatorname{tr} S_B = \operatorname{tr}[(C_B^{1/2}M_{B,t}C_B^{1/2})^{1/2}]$, but it need not be attained; the damped moment $\overline{M}_{B,t} = M_{B,t} + \lambda I$, $\lambda > 0$, gives a practical construction whose undamped value is bounded by the damped optimum. Damping or an approximate transform must therefore be charged at its achieved bound, rather than identified with $\mathcal{U}_t^\star$ for the original moments.

What WUSH improves in a nonlinear model. The route from transforms to convergence is explicit: $\mathcal{T}_t$ determines the curvature-weighted distortion D_t, which bounds the gradient perturbation during training and the loss change at export. Layer structure turns D_t into the bound $\mathcal{U}_t$, and WUSH minimizes that bound over the allowed input-side block transforms. When the structural curvature bound is exact, it minimizes D_t itself under the multiplicative model. Otherwise it minimizes a certified upper bound on D_t.

The activation covariance Σ_X of regression is replaced by C_B, and the weight moment becomes $M_{B,t} = \sum_j a_{B,j}w_{B,j,t}w_{B,j,t}^\top$; with unit factors $a_{B,j} = 1$, this is the diagonal block $(M_W)_{BB}$ of (4) at the current weights. Thus nonlinear output sensitivity weights the rows of the weight matrix, while the WUSH trace functional retains its form. The final equalizing rotation participates in the multiplicative optimum: without mixing, $P_B = I$ gives $\phi_{B,t} = \operatorname{tr}(S_B^2)$, whereas the optimum is $(\operatorname{tr} S_B)^2/g_B$. For a nonzero block, the optimum divided by the unmixed value lies in $[1/g_B, 1]$. A Hadamard matrix is a convenient equalizer (Remark E.4), not the unique optimal rotation.

The bounds now have the same rate-plus-residual structure as Theorem E.8. In (35), η proportional to $K^{-1/2}$ makes the transient and sampling terms of order $K^{-1/2}$, leaving the quantization contribution $\beta\overline{D}_K$. Under PL, (36) has a geometric transient, an $O(\eta)$ sampling term, and the quantization bound $(\beta+\mu)D_{\max}/(2\mu)$. These quantization coefficients are independent of the step size. The distortion budgets are evaluated along the actual trajectory, which can itself depend on η.

At fixed weight and curvature moments, $\mathcal{U}_t^\star$ is proportional to σ_W^2, and WUSH minimizes it. This gives a direct transform-selection objective for the quantization term in (47) and (48). Uniform residual bounds require the distortion moments to stay bounded; the theorem does not itself prove bounded network weights or a uniform curvature envelope for an arbitrary nonlinear architecture. Pointwise optimality at fixed weights also does not order the losses along two different training trajectories.