COLINEARITY DECAY: TRAINING QUANTIZATION-FRIENDLY TRANSFORMERS VIA DECOUPLED WEIGHT-SPACE REGULARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-bit quantization enables efficient Transformer deployment, yet activation outliers complicate fully quantized inference. Quantization methods act after or during training: post-training methods adapt pretrained models, while many training-time approaches address outliers through architectural changes, or by regularizing activations. Yet minimizing activation magnitude is incomplete: excessive suppression can degrade full-precision accuracy without consistently improving quantized models. We argue that quantization-friendly training should target harmful structural artifacts that cause extreme activations. To this end, we introduce Colinearity Decay (CD), a weight-space regularizer for matrix pairs within Transformer blocks. CD penalizes detrimental cross-matrix alignment and mitigates extreme activations. The decoupled update leaves architecture and task loss unchanged, adding little training overhead and no inference-time cost. Across ImageNet-1K pre-training, object detection, downstream fine-tuning, and small-scale LLM pre-training, CD improves quantized task performance across algorithms and bit widths while preserving unquantized performance. These results support decoupled weight-space regularization for training quantization-friendly Transformers.

1 INTRODUCTION

Transformers (Vaswani et al., 2017; Radford et al., 2019; Dosovitskiy et al., 2020; Liu et al., 2021) have become a standard architecture across computer vision and natural language processing. Their deployment cost, however, remains a practical obstacle when memory bandwidth, latency, or energy is limited. Low-bit quantization is a natural way to reduce this cost, but fully quantizing Transformers remains difficult. The main difficulty often comes from activations: a small number of channels or tokens can take much larger values than the rest, forcing coarse quantization scales and leaving the majority of activations represented with low effective precision.

Existing quantization methods for Transformers address this problem mostly at the quantization stage. Post-training quantization (PTQ) adapts pretrained models via calibration (Li et al., 2019; Wu et al., 2025) and equivalent reparameterization (Li et al., 2023; Ashkboos et al., 2024), while quantization-aware training (QAT) jointly optimizes model and quantization parameters under simulated low-bit operations (Esser et al., 2019; Liang et al., 2025a; Chen et al., 2025). These methods are effective, but mostly applied *after the full-precision model has already been trained*. The model itself may therefore remain poorly conditioned for low-bit quantization.

Another line of work reduces activation outliers *during training*. Architectural or optimization changes can mitigate attention sinks and other sources of extreme activations (Kaul et al., 2024; Sun et al., 2024; Agarwal et al., 2025; He et al., 2024; Park et al., 2025), while direct activation regularizers regularize activation distribution or magnitudes (Nrusimha et al., 2024; Liang et al., 2025b). Although these approaches operate through different mechanisms, they raise a common question for *quantization-friendly training*: should it suppress outliers as aggressively as possible, or instead *control the structural amplification that makes them harmful to low-bit deployment*?

In this paper, we argue that the latter strategy offers a better trade-off between full-precision and quantized accuracy. Direct activation-suppression methods such as TWEO (Liang et al., 2025b) are closely tied to FP8 training: reducing maximum activation values (e.g., to 30) can stabilize

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

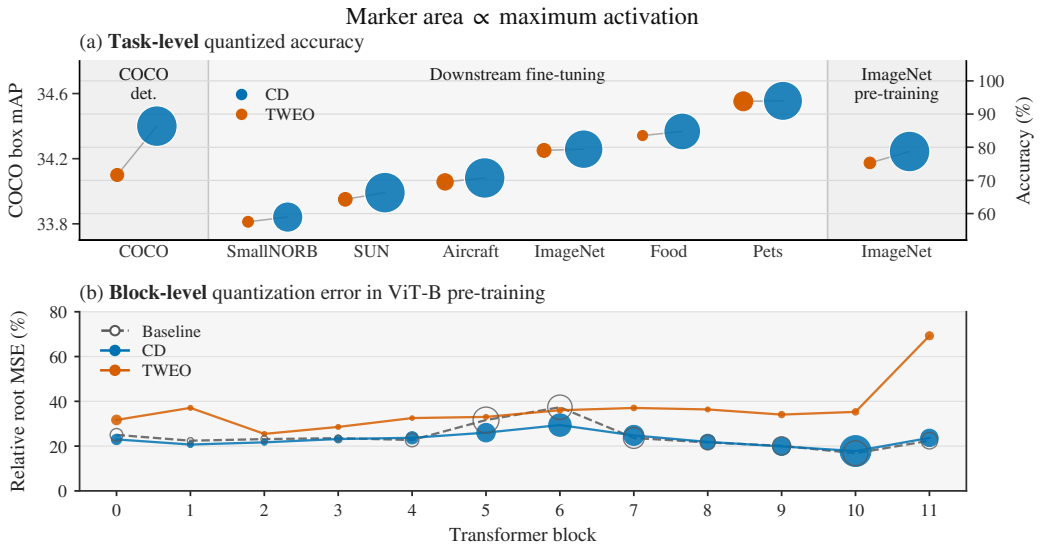

Figure 1: Smaller activations do not necessarily improve quantization. All results use RepQ-ViT (Li et al., 2023) W4A4; marker areas encode maximum activation. (a) CD attains higher quantized accuracy than TWEO despite larger activations. (b) At W_{F2}, where quantization error is largest, TWEO has higher RRMSE ($100\|y_q - y\|_2/\|y\|_2$) between quantized and full-precision outputs.

low-precision optimization on the FP8 hardware. But, these methods will potentially harm the accuracy of the trained FP models, and as a consequence harm the quantized models.

Figure 1 compares TWEO with our Colinearity Decay (CD) and provides task and block-level evidence for this argument. At the task level, TWEO strongly suppresses maximum activations, whereas CD retains larger activations but achieves higher quantized accuracy. At the block level, CD reduces error at the layers with the largest quantization errors, whereas TWEO amplifies error in nearly every block. These results do not imply that outliers are harmless. Rather, maximum activation is a useful diagnostic but not a sufficient training objective. Our key argument is that *quantization-friendly training should reduce harmful structural artifacts that cause extreme activations, but should not suppress activation magnitudes to a level that potentially degrades task performance.*

Figure 2 further reveals an essential practical advantage of our CD method: because CD is applied as a decoupled weight-space update (i.e., simply decay on parameters) rather than as an activation-regularization loss, it adds significantly smaller training overhead than TWEO.

CD is a decoupled weight-space regularizer for ordered matrix pairs in Transformer blocks. The method is motivated by the structural source of activation amplification: for a pair of adjacent or functionally adjacent matrices (W_1, W_2), large outputs can arise when rows of W_2 align with dominant left singular directions of W_1. Standard weight decay regularizes each matrix independently and does not directly penalize this cross-matrix alignment. CD instead penalizes the pairwise energy $\|W_2 W_1\|_F^2$, thus reducing large activation values (i.e., outliers).

Instead of adding an auxiliary loss to the training graph, CD applies a separate update in the spirit of decoupled weight decay (Loshchilov & Hutter, 2017) to the downstream matrix in each selected matrix pair. This update does not affect the task gradient. The pairing rule covers both composable pairs, such as normalization scales followed by projections, and functional pairs in attention and FFN sublayers. CD therefore keeps the architecture, task loss, and inference *intact*, adds little training-time matrix computation, and remains complementary to architectural outlier-mitigation methods.

We evaluate CD across vision tasks, including ImageNet-1K (Deng et al., 2009) pre-training, downstream fine-tuning, and COCO (Lin et al., 2014) detection, as well as small-scale LLM pre-training. Across these settings and a broad range of quantization algorithms and bit-width configurations, CD preserves unquantized task performance while improving its quantized counterpart. These gains are consistent across vision and language tasks and diverse Transformer architectures.

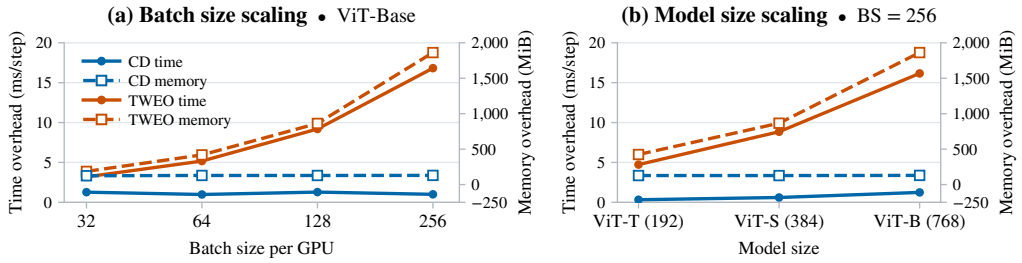

Figure 2: Training overhead relative to the baseline run on ImageNet-1K DDP pre-training. Left: time and memory overhead versus per-GPU batch size on ViT-B. Right: time and memory overhead versus model size at BS/GPU$= 256$. Time is in ms/step, and memory is in MiB.

Our contributions are as follows:

- We empirically show that excessive suppression of activation magnitudes can degrade both unquantized and quantized accuracy. Quantization-friendly training should instead target harmful structural artifacts that cause extreme activations.
- We introduce Colinearity Decay (CD) to reduce extreme cross-matrix alignment through decoupled weight-space regularization. CD leaves the model architecture, task loss, and inference computation unchanged, and adds little training overhead.
- We validate CD across ImageNet-1K pre-training, fine-tuning, COCO detection, and small-scale LLM pre-training. It preserves unquantized performance while improving its quantized counterpart across quantization algorithms and bit widths, as well as tasks and architectures.

2 Related Work

Quantization Methods for Transformers. Transformer quantization is complicated by activation outliers, which make low-bit activation quantization particularly difficult. For vision Transformers, representative approaches include simple percentile clipping (Li et al., 2019), scale reparameterization in RepQ-ViT (Li et al., 2023), reconstruction-based optimization in APHQ (Wu et al., 2025), and lightweight QAT in GPLQ (Liang et al., 2025a). For LLMs, weight-only PTQ includes RTN, GPTQ (Frantar et al., 2022), and AWQ (Lin et al., 2024), with the latter two using calibration activations; weight-activation methods include the reparameterization-based SmoothQuant (Xiao et al., 2023) and rotation-based QuaRot (Ashkboos et al., 2024). EfficientQAT (Chen et al., 2025) extends lightweight QAT to LLMs. CD complements these methods by reducing structural sources of extreme activations during training, making models easier to quantize across downstream pipelines.

Training-Time Outlier Mitigation. Another line of work mitigates outliers during training. Some methods (Kaul et al., 2024; Sun et al., 2024) connect outliers to attention sinks and introduce architectural changes, while others (Qiu et al., 2026; Park et al., 2025) modify the architecture or training recipe based on block design, or optimization. Activation regularization methods instead add training objectives: Nrusimha et al. (2024) combines QAT with kurtosis regularization of matrix outputs, whereas TWEO (Liang et al., 2025b) penalizes the absolute values of block outputs. CD instead applies decoupled weight-space regularization to ordered matrix pairs, leaving the model architecture and task loss unchanged while remaining compatible with architectural methods.

Outliers and Quantization Difficulty. The relation between outliers and quantization accuracy is more subtle than activation magnitude alone. TWEO (Liang et al., 2025b) shows that removing extreme activation outliers can enable FP8 training and improve quantization, whereas our experiments indicate that direct activation suppression is not always the best trade-off for quantized vision models. Related studies find that activation regularization can shift quantization difficulty to weights when unbalanced (Nrusimha et al., 2024), and that outlier metrics, including the max-to-mean ratio and kurtosis, do not reliably predict PTQ performance across optimizers (Vlassis et al., 2025). These observations are in line with our finding that addressing structural artifacts better preserves unquantized accuracy and improves downstream quantization than directly suppressing activation magnitudes.

3 METHOD

CD targets a structural source of activation outliers. In a Transformer block (Vaswani et al., 2017), large activation can be amplified not only by an individual matrix, but also by the alignment between two ordered matrices along the data flow. We first formalize this cross-matrix colinearity, then identify the relevant pairs in Transformer blocks, and finally give the decoupled CD update.

3.1 COLINEARITY AS A STRUCTURAL SOURCE OF OUTLIERS

Transformer activation outliers often emerge at the second FFN linear layer W_{F2} and propagate across blocks through skip connections (An et al., 2025). One prominent explanation links these outliers to attention heads representing a no-op (Bondarenko et al., 2023), motivating several modifications to the attention computation (Sun et al., 2024; Kaul et al., 2024). While these accounts explain outlier formation in terms of activation patterns, attention behavior, and a data-dependent view, we study a complementary source in the weights: *the colinearity between adjacent matrices along the data flow*.

TWEO (Liang et al., 2025b) finds that extreme outliers persist under random inputs but disappear when the weights are randomly initialized, indicating that they are mechanical artifacts arising from specific learned weight structures rather than input data. It further shows that the linearized expression $W_{F2}W_{F1}x$, obtained by omitting the activation function, captures the extreme coordinates of the actual FFN output. TWEO uses its findings to motivate systematic suppression through a direct penalty on block-output activations, whereas *we formulate the responsible cross-matrix structural amplification as an optimizable objective and regularize it directly*.

Following the decomposition in TWEO, we consider a general ordered pair (W_1, W_2), where $W_1 \in \mathbb{R}^{d_m \times d_{\text{in}}}$ is upstream and $W_2 \in \mathbb{R}^{d_{\text{out}} \times d_m}$ is downstream. Let $W_1 = USV^\top$ be its compact SVD. Here, $U = [u_1, \ldots, u_r]$ contains the left singular vectors and $S = \text{diag}(s_1, \ldots, s_r)$. For the j-th row $w_{2,j}^\top$ of W_2, the corresponding coordinate of the linearized output is

$$y_j = w_{2,j}^\top W_1 x = \sum_{i=1}^{r} s_i \left(w_{2,j}^\top u_i\right)\left(v_i^\top x\right). \tag{1}$$

Equation (1) directly explains this structural effect. If a row of W_2 aligns with a dominant left singular direction of W_1, its contribution is weighted by a large singular value, amplifying selected input components and potentially producing large activations even when neither matrix is unusual alone. We reproduce this behavior in outlier-heavy layers of ViT-B/16 and DINOv3 (Figure 4); extreme values in the linearized output often track those in the actual output. We further identify strong alignments through the inner products between $w_{2,j}$ and u_i, then prune them by zeroing the corresponding s_i. This intervention substantially reduces extreme activations and improves quantized accuracy. Appendix A.1 provides the intervention details and results, particularly Figure 5. Observations above motivate CD to suppress such harmful structural artifacts.

To formulate *colinearity* as an optimizable objective, we aggregate the squared singular-value-weighted projections across all directions and output rows. A simple transformation then gives an equivalent matrix-product form that avoids costly SVD computations:

$$\begin{aligned}\mathcal{L}_{\text{cd}}(W_1, W_2) &= \sum_{j=1}^{d_{\text{out}}} \sum_{i=1}^{r} \left(s_i w_{2,j}^\top u_i\right)^2 = \|W_2 US\|_F^2 \\ &= \|W_2 USV^\top\|_F^2 = \|W_2 W_1\|_F^2 .\end{aligned} \tag{2}$$

The objective is large when downstream rows place substantial mass on high-singular-value directions of the upstream matrix. Standard weight decay cannot penalize this cross-matrix alignment.

3.2 MATRIX PAIRS IN TRANSFORMER BLOCKS

We identify ordered matrix pairs in a standard pre-norm Transformer block (Radford et al., 2019):

$$X' = X + \text{Attn}(\text{LN}_1(X)), \qquad Y = X' + \text{FFN}(\text{LN}_2(X')). \tag{3}$$

Figure 3 summarizes these pairs and contrasts CD with activation regularization. An exact local matrix chain is a *composable pair*; matrices separated by a nonlinearity operator but connected along

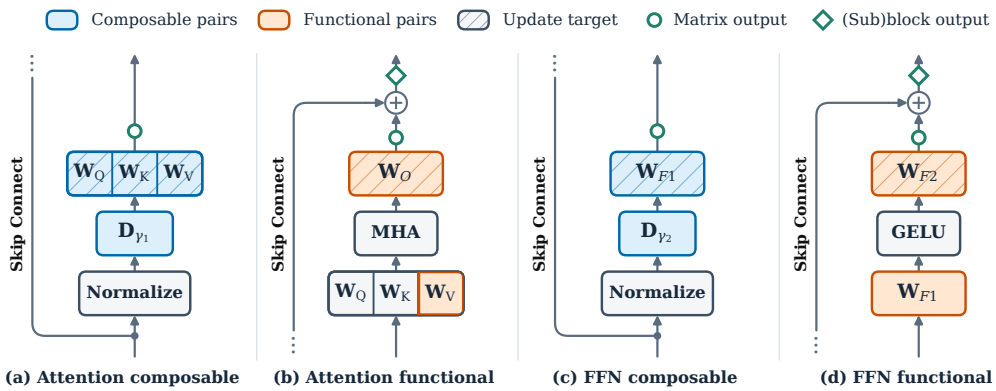

Figure 3: Ordered matrix pairs regularized by CD in a pre-norm Transformer block. Hatched (the latter) matrices receive the decoupled CD update. Circles and diamonds denote matrix and (sub)block outputs used by activation-regularization objectives, respectively; CD operates only on weights.

an amplification route form a *functional pair*. Activation regularization forms losses from matrix or (sub)block outputs, such as matrix-output kurtosis (Nrusimha et al., 2024) or the absolute values of block-output activations in TWEO (Liang et al., 2025b). CD is weight-only: it regularizes matrix pairs and updates the latter matrix, avoiding activation aggregation and backward propagation.

Composable pairs. The first case comes from normalization scales and the following projections. Omitting the bias term for clarity, LayerNorm or RMSNorm can be written as $\mathrm{LN}(X) = D_\gamma \hat{X}$, where $\hat{X}$ is the normalized activation and D_γ is the diagonal matrix induced by the scale vector γ. This exposes exact matrix pairs in both attention and FFN branches, shown in Fig. 3(a) and (c):

$$\begin{bmatrix} Q \\ K \\ V \end{bmatrix} = \begin{bmatrix} W_Q \\ W_K \\ W_V \end{bmatrix} D_{\gamma_1} \hat{X}_1 \,, \qquad H = W_{F1} D_{\gamma_2} \hat{X}_2 \,. \tag{4}$$

These pairs can generate outliers by the same mechanism as Eq. (1): a large scale can amplify the channel associated with a direction, producing a large output along that direction. This view connects to reparameterization methods in PTQ such as RepQ-ViT and SmoothQuant (Li et al., 2023; Xiao et al., 2023), which rescale adjacent activation-weight channels to shift quantization difficulty from per-tensor activations to per-channel weights. CD instead reduces this amplification during training.

Functional pairs. The second case is not always an exact matrix product, but the relevant matrices still form a route for outlier amplification. When the attention functional pair in Fig. 3(b) simply follows the single-head self-attention, the attention branch can be written as

$$A = \mathrm{softmax}\left(\frac{Q^\top K}{\sqrt{d_h}}\right) , \qquad \mathrm{Attn}(X) = W_O V A^\top = W_O W_V X A^\top . \tag{5}$$

Multi-head attention does not merge them into one dense product, but W_O still consumes the directions produced by the value branches. We therefore treat (W_V, W_O) as a functional pair.

The same idea applies to the FFN functional pair in Fig. 3(d). With ϕ denoting the activation function, the exact output and its linearized surrogate are

$$\mathrm{FFN}(x) = W_{F2}\, \phi(W_{F1} x) \,, \qquad \widehat{\mathrm{FFN}}(x) = W_{F2} W_{F1} x \,. \tag{6}$$

Although the intervening nonlinearity ϕ prevents the two matrices from forming an exact product, the linearized surrogate still tracks extreme coordinates in the actual FFN output, supporting (W_{F1}, W_{F2}) as a functional pair. Appendix A.1 and Figure 4 provide further analysis.

In this subsection, *we extend the ordered-matrix-pair formulation across Transformer blocks, covering composable and functional pairs in both attention and FFN branches*. Table 4(a) evaluates the two pair types individually and their combination, showing that the combination yields the best quantized accuracy while maintaining high full-precision accuracy.

3.3 COLINEARITY DECAY

For each selected pair, CD applies the pairwise objective in Eq. (2). The matrix gradients are

$$\frac{\partial \mathcal{L}_{\mathrm{cd}}}{\partial W_1} = 2W_2^\top W_2 W_1\,, \qquad \frac{\partial \mathcal{L}_{\mathrm{cd}}}{\partial W_2} = 2W_2 W_1 W_1^\top\,. \tag{7}$$

In practice, we apply CD only to the second matrix W_2 in each local pair. This avoids repeatedly decaying matrices that appear in two ordered matrix pairs. Using $W_1 = USV^\top$, the row-wise gradient with respect to the downstream matrix is $\partial \mathcal{L}_{\mathrm{cd}}/\partial w_{2,j}^\top = 2w_{2,j}^\top US^2U^\top$. Thus, CD shrinks downstream rows along high-singular-value directions of the upstream matrix, which are the directions most responsible for the amplification in Eq. (1). Since $US^2U^\top$ is positive semi-definite, the row-wise CD direction forms a non-obtuse angle (i.e., non-antagonistic) with the weight-decay direction.

Because CD is weight-only, its gradient depends on the paired weights rather than activations and has the simple closed form in Eq. (7). It can therefore *be computed without automatic differentiation and applied as a decoupled weight update in the style of AdamW weight decay* (Loshchilov & Hutter, 2017). Table 4(c) compares this implementation with adding CD to the training loss. The decoupled update yields higher full-precision and quantized accuracy and is our final implementation.

After computing the task gradient g_{task}, each optimization step proceeds from left to right as

$$\begin{aligned} W_2 \xrightarrow{\mathrm{CD}} W_2^{(\mathrm{c})} \xrightarrow{\mathrm{WD}} W_2^{(\mathrm{d})} \xrightarrow{\texttt{optimizer.step()}(g_{\mathrm{task}})} W_2^{+}, \\ W_2^{(\mathrm{c})} = W_2 - \eta\lambda_{\mathrm{cd}}\widetilde{D}_{\mathrm{cd}}, \qquad W_2^{(\mathrm{d})} = W_2^{(\mathrm{c})} - \eta\lambda_{\mathrm{wd}}W_2^{(\mathrm{c})}, \end{aligned} \tag{8}$$

where η is the learning rate, $D_{\mathrm{cd}} = W_2W_1W_1^\top$ is the raw CD direction, and $\widetilde{D}_{\mathrm{cd}}$ is its matrix-aligned version defined below. CD and WD are both applied directly to the weights, so their directions do not enter the optimizer states. *This implementation leaves the model architecture, task loss, and both training and inference computation graphs unchanged.*

Let λ_{f} be the baseline WD coefficient. CD replaces a portion λ_{cd}, leaving $\lambda_{\mathrm{wd}} = \lambda_{\mathrm{f}} - \lambda_{\mathrm{cd}}$ as residual WD. For each updated matrix, we scale the CD direction so that applying CD first and residual WD second matches the Frobenius norm obtained with baseline WD to first order. This dynamically regulates CD according to the WD strength and keeps individual weight-matrix Frobenius norms roughly matched to the baseline, isolating the effect of reducing colinearity between paired matrices.

We apply CD to all Transformer blocks. Composable pairs include the normalization scale with $[W_Q|W_K|W_V]$, and W_{F1}. Functional pairs include $W_V \to W_O$ and $W_{F1} \to W_{F2}$. In Swin (Liu et al., 2021), we include the normalization-scale/reduction pair in downsampling as composable. For SwiGLU-based gated MLPs (Shazeer, 2020), we design the proxy and derive a weight-only CD gradient for the down projection from the gate and up branches.

Detailed explanations and derivations of the above components are provided in Appendix A.2.

4 EXPERIMENTS

We evaluate CD on ImageNet-1K (Deng et al., 2009) pre-training, COCO (Lin et al., 2014) detection with joint detector training, downstream image classification fine-tuning (see Appendix A.7), and small-scale LLM pre-training. The results support two main observations:

- CD preserves full-precision accuracy while improving quantized accuracy across diverse algorithms and bit widths. These gains are consistent across tasks and architectures, spanning vision and language modeling, and make models better suited to low-bit deployment.
- Decoupled regularization of extreme structural alignments offers a better trade-off than activation regularization across training efficiency and full-precision and quantized accuracy.

4.1 EXPERIMENTAL SETUP

We train vision models with AMP (Micikevicius et al., 2017), train language models in BF16, and use the AdamW (Loshchilov & Hutter, 2017) optimizer. For each task, we evaluate quantization configurations. Within each matched comparison, methods share the architecture, base training recipe,

Table 1: ImageNet-1K pre-training results. All entries except Max. Act. are top-1 accuracy (%); WkAk denotes bit widths. **Boldface** marks the best and underlining the second-best per model.

Model	Method	Max. Act.	FP	RepQ-ViT W4A4	Percentile W4A6	APHQ W3A3	APHQ W4A4
ViT-B	Baseline	278.58	81.930	76.950	74.912	73.766	79.298
	TWEO	19.10	81.226	75.278	78.516	65.984	74.352
	KurtReg	41.95	80.466	75.576	74.144	69.156	75.930
	CD	233.99	**82.076**	**78.712**	**78.552**	**75.136**	**79.712**
ViT-B AttnBias	Baseline	171.43	81.836	77.140	75.756	77.140	80.446
	TWEO	13.38	81.262	73.408	78.744	69.968	76.426
	CD	130.93	**81.902**	**79.008**	**80.502**	**77.630**	**80.798**
Swin-S	Baseline	752.62	82.930	79.572	81.520	77.056	81.628
	TWEO	11.60	82.788	78.906	81.664	74.936	81.146
	CD	268.07	**83.162**	**80.580**	**82.170**	**77.596**	**81.966**
ViT-S	Baseline	118.55	79.924	68.452	72.680	64.192	75.086
	TWEO	18.83	79.852	68.304	73.268	62.486	74.872
	KurtReg	65.88	79.884	68.066	69.468	65.146	74.944
	CD	88.51	**79.956**	**70.872**	**75.978**	**65.720**	**75.162**
Swin-T	Baseline	431.77	80.918	75.188	78.396	74.966	79.406
	TWEO	16.86	81.056	75.872	**79.514**	74.404	79.312
	CD	135.43	**81.160**	**77.426**	79.482	**75.440**	**79.716**

and evaluation pipeline. For CD, we set λ_{cd} to one fifth of the baseline weight-decay coefficient and reduce λ_{wd} to four fifths of that coefficient, unless stated otherwise.

When reported, maximum activation is the largest value observed across module and Transformer-block outputs on the validation set. We treat this quantity as a diagnostic. Used alongside full-precision (FP) and quantized accuracy, it provides evidence for assessing the effects of activation suppression.

4.2 ImageNet-1K Pre-training

We follow the official DeiT (Touvron et al., 2021) and Swin (Liu et al., 2021) training recipes for ViT and Swin. Table 1 compares the baseline and CD with two activation-regularization methods. TWEO (Liang et al., 2025b) penalizes large Transformer-block outputs, whereas KurtReg uses the activation-kurtosis regularizer of Nrusimha et al. (2024), which is applied to linear-layer outputs. We keep the baseline settings fixed and add only each method's regularization terms and hyperparameters. Appendices A.3 and A.4 describe both objectives and report their ViT-B hyperparameter sweeps.

CD moderately reduces maximum activation by suppressing extreme structural amplification. Both activation-regularization methods reduce it much further, with TWEO yielding particularly small values. This stronger suppression, however, is accompanied by lower FP accuracy in most comparisons, and activation magnitude does not reliably predict quantized accuracy. TWEO consistently improves Percentile W4A6, but its gains are inconsistent with RepQ-ViT and APHQ, especially at W3A3; KurtReg is similarly inconsistent across quantizers. CD instead penalizes the weight-structural artifact that causes amplification, preserving FP accuracy while exceeding the matched baseline across the reported quantized settings. Appendix Table 6 confirms these gains across seeds, backbones, and quantizers. Results on our ViT-B adaptation of AttnBias (Agarwal et al., 2025) show that CD complements quantization-oriented architectural modification.

4.3 Efficiency

Figure 2 measures the time and memory overhead of CD and TWEO (Liang et al., 2025b) relative to matched baselines under the ImageNet-1K distributed data parallel pre-training setup. We evaluate batch-size scaling on ViT-B and model-size scaling across ViT-T/S/B at a fixed per-GPU batch size of 256. CD incurs substantially lower overhead than TWEO under both scaling settings.

Table 2: Detection results on COCO. We report $\mathrm{mAP}^{\mathrm{box}}/\mathrm{mAP}^{\mathrm{mask}}$ in percent. All quantization results use W4A4 settings. Max. Act. is the maximum backbone activation.

(a) Method comparison on Swin-T 1x.

Metric	Baseline	Noise	TWEO	CD
Max. Act.	191.93	224.66	15.27	149.17
FP mAP	42.6 / 39.3	42.5 / 39.2	41.0 / 37.9	**42.9 / 39.6**
RepQ-ViT	31.1 / 30.2	32.8 / 31.8	34.1 / 32.6	**34.4 / 33.0**

(b) Results across schedules, backbones, and quantization algorithms.

	Swin-T 1x		Swin-T 3x		Swin-S 3x	
Metric	Base	CD	Base	CD	Base	CD
FP mAP	42.6 / 39.3	**42.9 / 39.6**	45.9 / 41.5	**46.0 / 41.6**	48.3 / 43.2	**48.4 / 43.2**
Percentile	25.9 / 25.6	**30.4 / 29.6**	32.6 / 31.9	**33.6 / 32.4**	39.7 / 37.6	**40.7 / 38.1**
RepQ-ViT	31.1 / 30.2	**34.4 / 33.0**	36.6 / 35.6	**37.5 / 36.1**	41.4 / 38.9	**42.4 / 39.6**
APHQ	37.9 / 35.9	**38.1 / 36.2**	39.0 / 38.0	**40.2 / 38.5**	40.6 / 39.2	**42.5 / 40.4**
GPLQ	39.5 / 36.8	**39.9 / 37.3**	41.7 / 39.0	**42.1 / 39.1**	44.5 / 40.7	**44.9 / 40.9**

In panel (a), larger per-GPU batches increase the activation tensors processed per update and hence the memory and computation required by TWEO's activation regularization. CD is a weight-only update, so its overhead stays nearly constant at approximately 1 ms/step and 128 MiB. Panel (b) shows a similar scaling trend with model size. Larger hidden dimensions enlarge the activations used by TWEO, causing its time and memory overhead to increase from ViT-T to ViT-B. CD instead operates on weights already resident in GPU memory. Its peak-memory increase remains near 128 MiB, while its time overhead remains only 0.32–1.24 ms/step across model sizes.

4.4 COCO Detection

We next study detector training in MMDetection (Chen et al., 2019) with Mask R-CNN (He et al., 2017) and Swin (Liu et al., 2021) backbones. We initialize the backbone from Swin checkpoints and train the backbone and detector jointly. CD, TWEO (Liang et al., 2025b), and the Noise (Baskin et al., 2021) (uniform-noise injection in weights) baseline are applied only to the backbone.

Panel (a) shows that CD substantially improves the quantized detector while preserving FP accuracy. The pattern is consistent with classification pre-training: the Noise and TWEO baselines also improve quantized accuracy but yield a worse trade-off than CD, with lower FP accuracy.

Panel (b) of Table 2 extends this comparison across training schedules, model sizes, and quantization algorithms. The gains remain visible for the simple percentile (Li et al., 2019) PTQ baseline, reparameterization-based RepQ-ViT (Li et al., 2023), reconstruction-based APHQ (Wu et al., 2025), and QAT baseline GPLQ (Liang et al., 2025a). Across the tested quantized settings, CD consistently improves both box and mask mAP, with clear gains for simpler quantization methods. Additional notes comparing our reproduced checkpoints with official results are given in Appendix A.6.

4.5 LLM Pre-training

We test whether CD transfers beyond vision using GPT-2 Medium and Large (Radford et al., 2019) and a 1B ModernLM equipped with architectural features common in modern LLMs, trained on 30B or 60B FineWeb-Edu tokens (Penedo et al., 2024). We report validation bits per byte (BPB; lower is better) in BF16 and under RTN, GPTQ (Frantar et al., 2022), AWQ (Lin et al., 2024), SmoothQuant (Xiao et al., 2023), and QuaRot (Ashkboos et al., 2024). Table 3 reports the results; Appendix A.8 provides the architecture, training, and quantization details.

CD lowers BF16 BPB in four of five comparisons; the remaining ModernLM 30B result differs by only 1e-4. Under quantization, CD generally yields comparable or lower BPB across the evaluated methods and settings. The reductions are most visible under RTN, particularly for GPT-2, and appear under both weight-only and activation quantization. These results provide preliminary evidence that CD transfers beyond vision tasks and ViT-style architectures.

Table 3: Validation BPB for LLM trained on FineWeb-Edu subsets; lower is better. Bold values mark the lower BPB within each matched baseline/CD pair.

Model		GPT-2 M (355M)		GPT-2 Large (774M)				ModernLM (1B)			
Tokens		30B		30B		60B		30B		60B	
Method		Baseline	CD	Baseline	CD	Baseline	CD	Baseline	CD	Baseline	CD
BF16	W16A16	0.8550	**0.8493**	0.8106	**0.8090**	0.7898	**0.7888**	**0.7732**	0.7733	0.7565	**0.7560**
RTN	W4A16	1.1097	**0.8928**	0.9394	**0.8374**	0.8754	**0.8332**	0.7913	**0.7892**	0.7924	**0.7825**
GPTQ	W4A16	0.8753	**0.8714**	0.8253	**0.8225**	0.8117	**0.8105**	0.7841	**0.7831**	0.7795	**0.7711**
AWQ	W4A16	0.8677	**0.8610**	0.8230	**0.8168**	0.8011	**0.7994**	**0.7798**	**0.7798**	0.7662	**0.7650**
SQ	W8A8	0.8565	**0.8506**	0.8154	**0.8098**	0.8130	**0.7900**	0.7749	**0.7742**	0.7608	**0.7575**
QuaRot	W4A6	0.8751	**0.8643**	0.8241	**0.8180**	0.8041	**0.8020**	0.7847	**0.7845**	0.7780	**0.7709**

Table 4: Ablations on ViT-B pre-training. Max. Act. is the largest module/block output; FP and W4A4 are top-1 accuracies (%). W4A4 uses RepQ-ViT. Panel (d) compares added WD and CD.

(a) Matrix-pair selection.

Pairs	Max. Act.	FP	W4A4
Baseline	278.58	81.930	76.950
Composable	286.14	81.860	77.924
Functional	189.95	**82.084**	77.342
Both (CD)	233.99	82.076	**78.712**

(b) CD strength.

λ_{cd}	Max. Act.	FP	W4A4
0.000	278.58	81.930	76.950
0.005	203.52	81.710	78.510
0.010	233.99	**82.076**	**78.712**
0.015	196.56	81.872	78.282

(c) Update form.

Method	Max. Act.	FP	W4A4
Baseline	278.58	81.930	76.950
Decoupled CD	233.99	**82.076**	**78.712**
CD loss, 0.0025	202.61	81.724	76.270
CD loss, 0.005	177.74	81.778	76.576
CD loss, 0.010	104.99	81.476	76.204

(d) Additional regularization.

Method	Max. Act.	FP	W4A4
Baseline	278.58	81.930	76.950
WD +0.010	176.56	81.814	75.726
CD +0.010	164.36	**81.874**	**78.310**
WD +0.015	248.77	**82.212**	75.574
CD +0.015	147.93	81.926	**78.630**

4.6 Ablations

We report four ablations on ViT-B pre-training in Table 4. Panels (a)–(c) motivate our default design. Combining composable and functional pairs in panel (a) gives the best RepQ-ViT (Li et al., 2023) accuracy while maintaining FP accuracy. Panel (b) identifies $\lambda_{\text{cd}} = 0.01$ as the tested strength yielding the highest FP and RepQ-ViT accuracy, and panel (c) shows that the decoupled update outperforms the CD-loss variants on both metrics. These choices balance FP and quantized accuracy.

Panel (d) compares adding CD with increasing WD by the same coefficient. At both coefficients, additional CD improves RepQ-ViT accuracy over the baseline, whereas additional WD slightly lowers it. This indicates that CD provides a regularization distinct from WD within matched comparisons.

5 Conclusion

We studied cross-matrix colinearity in quantization-friendly Transformer training and introduced Colinearity Decay (CD), a decoupled weight-space regularizer for matrix pairs. Rather than directly minimizing activation magnitudes, CD penalized harmful cross-matrix alignment, a structural artifact that amplified extreme activations. Its decoupled update left architecture, task loss, and inference unchanged, added little training overhead, and remained complementary to existing quantization and outlier-mitigation methods. Across ImageNet-1K pre-training, downstream fine-tuning, COCO detection, and small-scale LLM pre-training, CD preserved unquantized task performance while improving its quantized counterpart across quantization algorithms and bit widths. These gains held across vision and language tasks and diverse Transformer architectures. Overall, these results support cross-matrix colinearity as a useful target for quantization-friendly Transformer training.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation, the implementation of parts of the research code, and the review of mathematical claims and derivations for correctness. We did not use generative AI tools to propose or refine the research hypotheses, design or provide feedback on the research methodology or experiments, or interpret the experimental results. The development of theoretical models or conceptual frameworks, generation of synthetic datasets, development or writing of formal mathematical proofs, dataset cleaning or reformatting, and qualitative or thematic data analysis were not applicable to this work. Additionally, we used generative AI tools to create or modify scientific figures, edit software code, source or search for information, draft parts of the manuscript, and edit the manuscript to improve readability. The authors carefully reviewed all AI-assisted text, figures, and mathematical content, and inspected and tested all AI-assisted code. The authors take full responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work uses public benchmark datasets, standard model families, and conventional machine-learning training and evaluation procedures. It does not involve human subjects or private personal data, nor does it release high-risk data or models.

REPRODUCIBILITY STATEMENT

Section 3 specifies the CD objective, matrix-pair construction, normalization, and decoupled update. Appendix A.2 provides detailed implementation rules and derivations. Section 4 reports the main training, quantization, and evaluation settings for each task, while Appendices A.3–A.9 provide additional experimental configurations and reproduction details. All experiments use public datasets, publicly available model families and checkpoints, and established evaluation methods.

REFERENCES

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. Systematic outliers in large language models. In *International Conference on Learning Representations*, volume 2025, pp. 31483–31504, 2025.

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. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024.

Chaim Baskin, Evgenii Zheltonozhkii, Tal Rozen, Natan Liss, Yoav Chai, Eli Schwartz, Raja Giryes, Alexander M Bronstein, and Avi Mendelson. Nice: Noise injection and clamping estimation for neural network quantization. *Mathematics*, 9(17):2144, 2021.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Quantizable transformers: Removing outliers by helping attention heads do nothing. *Advances in Neural Information Processing Systems*, 36:75067–75096, 2023.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. EfficientQAT: Efficient quantization-aware training for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10081–10100, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Bobby He, Lorenzo Noci, Daniele Paliotta, Imanol Schlag, and Thomas Hofmann. Understanding and minimising outlier features in transformer training. *Advances in Neural Information Processing Systems*, 37:83786–83846, 2024.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4246–4253, 2020.

Andrej Karpathy. nanochat: The best chatgpt that $100 can buy, 2025. URL `https://github.com/karpathy/nanochat`.

Prannay Kaul, Chengcheng Ma, Ismail Elezi, and Jiankang Deng. From attention to activation: Unravelling the enigmas of large language models. *arXiv preprint arXiv:2410.17174*, 2024.

Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pp. II–104. IEEE, 2004.

Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan. Fully quantized network for object detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2805–2814. IEEE, 2019.

Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. Repq-vit: Scale reparameterization for post-training quantization of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17227–17236, 2023.

Guang Liang, Xinyao Liu, and Jianxin Wu. Gplq: A general, practical, and lightning qat method for vision transformers. *arXiv preprint arXiv:2506.11784*, 2025a.

Guang Liang, Jie Shao, Ningyuan Tang, Xinyao Liu, and Jianxin Wu. Tweo: Transformers without extreme outliers enables fp8 training and quantization for dummies. *arXiv preprint arXiv:2511.23225*, 2025b.

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. *Proceedings of machine learning and systems*, 6: 87–100, 2024.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

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.

Aniruddha Nrusimha, Mayank Mishra, Naigang Wang, Dan Alistarh, Rameswar Panda, and Yoon Kim. Mitigating the impact of outlier channels for language model quantization with activation regularization. *arXiv preprint arXiv:2404.03605*, 2024.

Jungwoo Park, Taewhoo Lee, Chanwoong Yoon, Hyeon Hwang, and Jaewoo Kang. Outlier-safe pre-training for robust 4-bit quantization of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12582–12600, 2025.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Zihan Qiu, Zeyu Huang, Kaiyue Wen, Peng Jin, Bo Zheng, Yuxin Zhou, Haofeng Huang, Zekun Wang, Xiao Li, Huaqing Zhang, et al. A unified view of attention and residual sinks: Outlier-driven rescaling is essential for transformer training. *arXiv preprint arXiv:2601.22966*, 2026.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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

Yikang Shen, Matthew Stallone, Mayank Mishra, Gaoyuan Zhang, Shawn Tan, Aditya Prasad, Adriana Meza Soria, David D Cox, and Rameswar Panda. Power scheduler: A batch size and token number agnostic learning rate scheduler. *arXiv preprint arXiv:2408.13359*, 2024.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*, 2024.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Georgios Vlassis, Saleh Ashkboos, Alexandra Volkova, Torsten Hoefler, and Dan Alistarh. Beyond outliers: A study of optimizers under quantization. *arXiv preprint arXiv:2509.23500*, 2025.

Zhuguanyu Wu, Jiayi Zhang, Jiaxin Chen, Jinyang Guo, Di Huang, and Yunhong Wang. Aphq-vit: Post-training quantization with average perturbation hessian based reconstruction for vision transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9686–9695, 2025.

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*, pp. 38087–38099. PMLR, 2023.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

A Appendix

A.1 FFN Outliers and FC1-FC2 Colinearity

We first examine whether the linear surrogate in Eq. (6) captures the large coordinates produced by the actual FFN branch. Figure 4 compares the real FFN branch output with the no-activation approximation $W_{F2}W_{F1}x$ on selected blocks of ViT-B/16 and DINOv3 (Siméoni et al., 2025) ViT-B/16. Although the surrogate ignores the activation function, the large-magnitude outputs often follow the same directions as the real branch output. This agreement indicates that the matrix pair (W_{F1}, W_{F2}) already contains a structural source of FFN outliers.

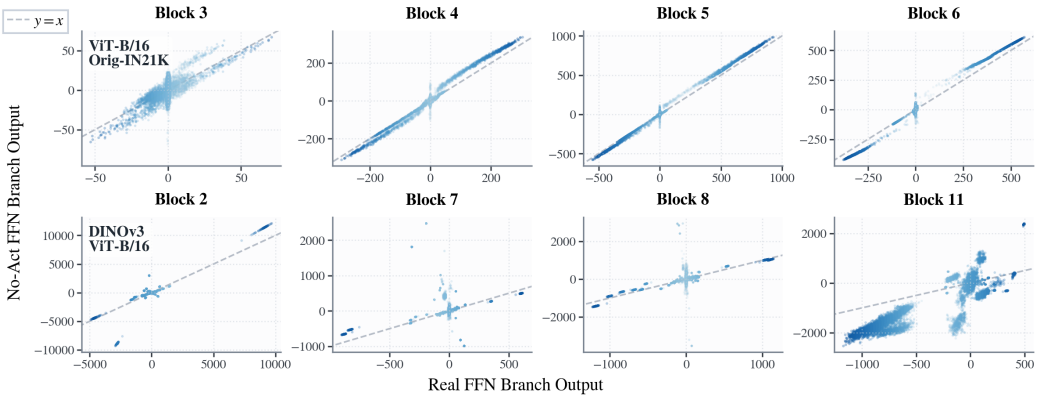

Figure 4: Real FFN branch output versus the no-activation surrogate $W_{F2}W_{F1}x$ across selected blocks of ViT-B/16 and DINOv3 ViT-B/16. The dashed line denotes $y = x$.

Eq. (1) gives a more specific explanation of this effect. Let $W_{F1} = USV^\top$, and let $\hat{w}_{2,j}$ denote the j-th row of W_{F2} after row normalization. We measure the alignment between the i-th left singular direction of W_{F1} and the rows of W_{F2} by

$$\alpha_{i,j} = |\hat{w}_{2,j}^\top u_i|. \tag{9}$$

Figure 5 visualizes this alignment in outlier-heavy layers. Blocks 4, 5, and 6 contain a small number of singular-vector directions whose maximum row alignment is far above the bulk of the distribution. These peaks are not spread uniformly across all directions; they concentrate on a few W_{F1} left singular vectors. This pattern is consistent with the colinearity mechanism in Eq. (1): the outlier is induced by a specific cross-matrix alignment rather than by a uniformly large W_{F2} row norm.

To test whether these directions are redundant, we aggregate the alignment scores over the selected layers, sort the strongest directions, and sequentially set the corresponding FC1 singular values to zero. The three removed directions are Block 5 index 1, Block 6 index 1, and Block 4 index 9. Removing the top one or two directions substantially reduces the maximum activation and improves RepQ-ViT (Li et al., 2023) W4A4 accuracy while leaving FP accuracy close to the baseline.

This diagnostic intervention supports two observations. First, FFN outliers in ViT are tightly connected to the colinearity between dominant W_{F1} singular directions and selected W_{F2} rows.

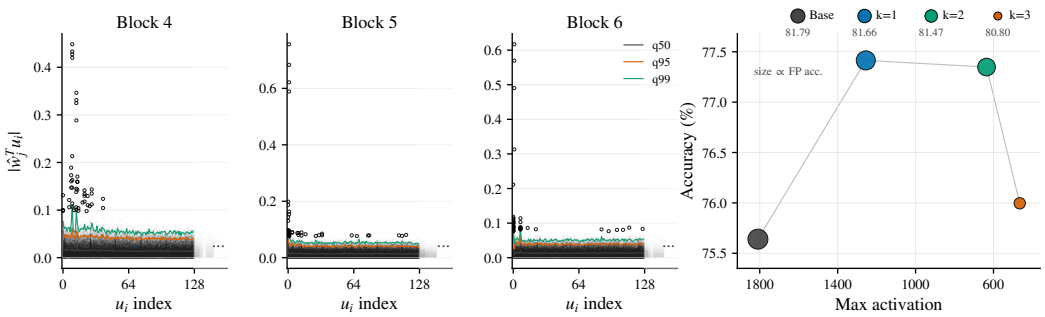

Figure 5: W_{F1}-W_{F2} alignment and its effect on quantization. Left: normalized alignment $|\hat{w}_{2,j}^\top u_i|$ between W_{F1} left singular vectors and W_{F2} rows in Blocks 4, 5, and 6. Right: effect of sequentially removing the most aligned singular directions on maximum activation and RepQ-ViT W4A4 accuracy. Marker area is proportional to FP accuracy. All conducted on `vit_base.orig_in21k_ft_in1k`

Second, part of this colinearity is redundant for the FP model. Controlling this harmful alignment can reduce outliers and improve the quantized model, which is the mechanism targeted by CD.

A.2 Method Derivations and Details

We collect the derivations and implementation details omitted from the main text.

CD gradient derivation. For the CD objective,

$$\mathcal{L}_{\text{cd}}(W_1, W_2) = \|W_2 W_1\|_F^2 = \text{tr}(W_1^\top W_2^\top W_2 W_1), \tag{10}$$

where we use $\|A\|_F^2 = \text{tr}(A^\top A)$ and the cyclic property of the trace. We compute gradients through differentials and use the convention $\text{d}\mathcal{L} = \text{tr}((\partial\mathcal{L}/\partial W)^\top \text{d}W)$. First fix W_2 and set $A = W_2^\top W_2$. Since $A = A^\top$,

$$\begin{aligned} \text{d}_{W_1}\mathcal{L}_{\text{cd}} &= \text{tr}(\text{d}W_1^\top A W_1) + \text{tr}(W_1^\top A\, \text{d}W_1) \\ &= \text{tr}((AW_1)^\top \text{d}W_1) + \text{tr}((A^\top W_1)^\top \text{d}W_1) \\ &= \text{tr}((2AW_1)^\top \text{d}W_1). \end{aligned} \tag{11}$$

Thus $\partial\mathcal{L}_{\text{cd}}/\partial W_1 = 2W_2^\top W_2 W_1$. Next fix W_1 and set $B = W_1 W_1^\top$. By cyclicity, $\mathcal{L}_{\text{cd}} = \text{tr}(W_2 B W_2^\top)$, with $B = B^\top$. Therefore,

$$\begin{aligned} \text{d}_{W_2}\mathcal{L}_{\text{cd}} &= \text{tr}(\text{d}W_2 B W_2^\top) + \text{tr}(W_2 B\, \text{d}W_2^\top) \\ &= \text{tr}((W_2 B^\top)^\top \text{d}W_2) + \text{tr}((W_2 B)^\top \text{d}W_2) \\ &= \text{tr}((2W_2 B)^\top \text{d}W_2). \end{aligned} \tag{12}$$

The gradients are therefore

$$\frac{\partial\mathcal{L}_{\text{cd}}}{\partial W_1} = 2W_2^\top W_2 W_1, \qquad \frac{\partial\mathcal{L}_{\text{cd}}}{\partial W_2} = 2W_2 W_1 W_1^\top, \tag{13}$$

which matches Eq. (7). Using $W_1 = USV^\top$ gives $W_1 W_1^\top = US^2U^\top$, so the j-th downstream row gradient is $2w_{2,j}^\top US^2U^\top$. This shows that CD removes the component of $w_{2,j}$ in the left singular subspace of W_1, with larger shrinkage on directions with larger singular values.

Directional compatibility with weight decay. We next show that, for a downstream row vector, the CD gradient and the weight-decay gradient form an angle no larger than $90°$. Let

$$M = US^2U^\top \succeq 0, \qquad g_{\text{cd},j} = 2w_{2,j}^\top M, \qquad g_{\text{wd},j} = w_{2,j}^\top, \tag{14}$$

where $g_{\text{cd},j}$ is the row-wise CD gradient and $g_{\text{wd},j}$ is the row-wise gradient of the weight-decay term $\frac{1}{2}\|W_2\|_F^2$. The matrix M is positive semi-definite because, for any vector z,

$$z^\top M z = z^\top US^2U^\top z = \|SU^\top z\|_2^2 \geq 0. \tag{15}$$

Using the Euclidean inner product between row gradients, we obtain

$$\langle g_{\mathrm{cd},j}, g_{\mathrm{wd},j} \rangle = 2 w_{2,j}^{\top} M w_{2,j} \geq 0. \tag{16}$$

When both gradients are nonzero, their cosine similarity is therefore nonnegative:

$$\cos\theta_j = \frac{\langle g_{\mathrm{cd},j}, g_{\mathrm{wd},j} \rangle}{\|g_{\mathrm{cd},j}\|_2 \|g_{\mathrm{wd},j}\|_2} \geq 0. \tag{17}$$

Hence $\theta_j \in [0^\circ, 90^\circ]$. If either gradient is zero, the angle is not defined, but CD still has no component opposing the weight-decay gradient.

Matrix-aligned CD and regularization budget. Matrix-aligned CD replaces part of the baseline WD while matching its radial shrinkage for each updated matrix. Let W denote the current downstream matrix and D its raw CD direction, with the factor 2 absorbed into the CD coefficient. We write the baseline WD coefficient as λ_{f}, the residual WD coefficient as λ_{wd}, and the replaced portion as $\lambda_{\mathrm{cd}} = \lambda_{\mathrm{f}} - \lambda_{\mathrm{wd}}$. Define

$$N = \|W\|_F^2, \qquad B = \langle W, D \rangle_F, \qquad A = \|D\|_F^2. \tag{18}$$

For a CD displacement sD, let $a = 1 - \eta\lambda_{\mathrm{wd}}$ and $b = 1 - \eta\lambda_{\mathrm{f}}$. Exact Frobenius-norm matching between CD followed by residual WD and full WD alone requires

$$\|a(W - sD)\|_F = \|bW\|_F. \tag{19}$$

Squaring Eq. (19) yields

$$As^2 - 2Bs + C = 0, \qquad C = \left[1 - \left(\frac{b}{a}\right)^2\right] N. \tag{20}$$

For $A > 0$ and $B > 0$, sufficiently small η gives two real roots,

$$s_{\pm} = \frac{B \pm \sqrt{B^2 - AC}}{A}. \tag{21}$$

As $\eta \to 0$, we have $C \to 0$, so $s_- \to 0$ whereas $s_+ \to 2B/A$. We choose s_- because the regularization step must vanish with the learning rate. Rationalizing its numerator avoids subtracting two nearly equal quantities and gives the equivalent form

$$s_{\mathrm{exact}} = \frac{C}{B + \sqrt{B^2 - AC}}. \tag{22}$$

We use a first-order approximation that avoids computing A and the square root. Since $b = a - \eta\lambda_{\mathrm{cd}}$,

$$C = \left[\frac{2\eta\lambda_{\mathrm{cd}}}{a} - \left(\frac{\eta\lambda_{\mathrm{cd}}}{a}\right)^2\right] N = 2\eta\lambda_{\mathrm{cd}} N + O(\eta^2). \tag{23}$$

At the current update, A and B are fixed with respect to η. Since $C = O(\eta)$ and $B > 0$, a first-order expansion gives

$$\sqrt{B^2 - AC} = B\sqrt{1 - \frac{AC}{B^2}} = B - \frac{AC}{2B} + O(\eta^2). \tag{24}$$

The denominator of Eq. (22) is thus $2B + O(\eta)$, and the selected root satisfies

$$s_{\mathrm{exact}} = \eta\lambda_{\mathrm{cd}} \frac{N}{B} + O(\eta^2). \tag{25}$$

Thus, for $B > 0$, we set

$$k = \frac{N}{B}, \qquad \widetilde{D} = kD, \qquad W \leftarrow W - \eta\lambda_{\mathrm{cd}}\widetilde{D}, \tag{26}$$

and use $k = 0$ when $B = 0$. Because $\langle W, \widetilde{D} \rangle_F = N$, the CD step produces the same first-order change in squared Frobenius norm as WD with coefficient λ_{cd}. Applying residual WD afterward therefore matches full WD to first order in η; the discrepancy consists only of second-order terms

from the CD displacement and its interaction with residual WD. This alignment requires only the two scalar reductions N and B and adds no matrix multiplication beyond the raw CD direction.

As an empirical check, we compare the final checkpoints of the ViT-B baseline and CD runs. Across the 12 blocks, the median blockwise CD-to-baseline ratios of $\|W_{F1}\|_F$, $\|W_{F2}\|_F$, and $\|W_{F2}W_{F1}\|_F$ are 1.00, 1.00, and 0.33, respectively. Matrix-aligned CD therefore preserves the individual matrix scales while reducing their pairwise product energy significantly.

SwiGLU down-projection CD gradient. Consider a bias-free SwiGLU layer with gate, up, and down projections $G, U \in \mathbb{R}^{m\times d}$ and $P \in \mathbb{R}^{o\times m}$:

$$h(x) = \mathrm{SiLU}(Gx) \odot (Ux), \qquad y(x) = Ph(x). \tag{27}$$

Let $g_k^\top$ and $u_k^\top$ denote the k-th rows of G and U. For a gate channel in its positive, approximately linear regime, $\mathrm{SiLU}(g_k^\top x) \approx g_k^\top x$, and hence

$$h_k(x) \approx (g_k^\top x)(u_k^\top x) = x^\top S_k x, \qquad S_k = \frac{1}{2}\left(g_k u_k^\top + u_k g_k^\top\right). \tag{28}$$

Applying the down projection aggregates these quadratic forms across the m intermediate channels. The j-th output coordinate becomes

$$y_j(x) = \sum_{k=1}^{m} P_{jk} h_k(x) \approx \sum_{k=1}^{m} P_{jk} x^\top S_k x = x^\top A_j x, \qquad A_j = \sum_{k=1}^{m} P_{jk} S_k. \tag{29}$$

Since $|x^\top A_j x| \le \|x\|_2^2 \|A_j\|_F$, we use the following weight-only proxy for post-down structural amplification:

$$\mathcal{L}_{\mathrm{cd}}^{\mathrm{SwiGLU}}(G,U,P) = \sum_{j=1}^{o} \left\| \sum_{k=1}^{m} P_{jk} S_k \right\|_F^2 = \|P\mathcal{B}\|_F^2, \tag{30}$$

where the k-th row of $\mathcal{B} \in \mathbb{R}^{m\times d^2}$ is $\mathrm{vec}(S_k)^\top$. We do not construct this d^2-dimensional operator. Its Gram matrix $K = \mathcal{B}\mathcal{B}^\top \in \mathbb{R}^{m\times m}$ has the closed form

$$K = \frac{1}{2}\left[(GG^\top) \odot (UU^\top) + (GU^\top) \odot (UG^\top)\right] \succeq 0. \tag{31}$$

Consequently,

$$\mathcal{L}_{\mathrm{cd}}^{\mathrm{SwiGLU}} = \mathrm{tr}(PKP^\top), \qquad \frac{\partial \mathcal{L}_{\mathrm{cd}}^{\mathrm{SwiGLU}}}{\partial P} = 2PK. \tag{32}$$

The gradient depends only on the three projection matrices. It also satisfies

$$\langle P, 2PK\rangle_F = 2\,\mathrm{tr}(PKP^\top) = 2\mathcal{L}_{\mathrm{cd}}^{\mathrm{SwiGLU}} \ge 0, \tag{33}$$

so it does not oppose the weight-decay direction. Absorbing the factor 2 into λ_{cd} gives the raw down-projection CD direction $D = PK$. The matrix-aligned update in Eq. (26) then applies directly with $W = P$ and $D = PK$. Because this update changes only P, it controls post-down amplification rather than the pre-down activation $h(x)$ at the current step.

A.3 TWEO Reproduction and Hyperparameter Sweep

TWEO (Liang et al., 2025b) is an activation-regularization method designed to suppress extreme activation outliers during training. Unlike CD, which is implemented as a decoupled update on matrix pairs, TWEO regularizes block-output activations directly through the training graph.

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{task}} + \lambda(t)\mathcal{L}_{\mathrm{TWEO}}, \qquad \mathcal{L}_{\mathrm{TWEO}} = \frac{1}{L}\sum_{l=1}^{L} \mathbb{E}\left[\left(\frac{|A^{(l)}|}{\tau+\epsilon}\right)^p\right]. \tag{34}$$

For detection and fine-tuning, we use the rescaled objective

$$\mathcal{L}_{\mathrm{total}}^{\mathrm{rescaled}} = \mathcal{L}_{\mathrm{task}} + \lambda(t)\,\mathrm{sg}\left(\frac{\mathcal{L}_{\mathrm{task}}}{\mathcal{L}_{\mathrm{TWEO}}}\right)\mathcal{L}_{\mathrm{TWEO}}, \tag{35}$$

Table 5: Hyperparameter sweeps for TWEO and KurtReg on ViT-B ImageNet-1K pre-training. Entries are top-1 accuracy (%); W4A4 uses RepQ-ViT. Bold marks the configurations used in Table 1.

(a) TWEO

$\lambda_{\max}$	$\lambda_{\min}$	τ	p	FP	W4A4
10^{-1}	10^{-3}	3	4	80.440	73.824
10^{-2}	10^{-2}	**3**	**4**	**81.226**	**75.278**
10^{-2}	10^{-3}	3	4	81.144	74.296
10^{-2}	10^{-3}	3	2	80.014	71.812
10^{-3}	10^{-4}	3	4	80.300	74.104

(b) KurtReg

λ_{Kurt}	FP	W4A4
2×10^{-5}	79.730	75.500
1×10^{-5}	80.312	74.976
5×10^{-6}	**80.466**	**75.576**
2×10^{-6}	81.250	73.762

where $A^{(l)}$ denotes the final output activation of the l-th Transformer block, $\lambda(t)$ is the step-dependent loss weight, τ is a magnitude scale, and p is the power controlling how strongly large activations are penalized. The operator sg stops gradients through the loss ratio, which scales the TWEO term to the current task-loss magnitude before $\lambda(t)$ controls its weight. In the original paper, TWEO is applied to every block output, typically with $p = 4$, $\tau = 3$, and an optional cosine schedule on $\lambda(t)$.

For ImageNet-1K pre-training, we implement this objective in the same codebase and training recipe as the baseline and CD. Table 5(a) varies the endpoints $\lambda_{\max}$ and $\lambda_{\min}$ of the regularization schedule, and the power p on ViT-B. The selected setting attains the highest FP and RepQ-ViT W4A4 accuracy among the tested configurations and produces the TWEO entry in Table 1.

When we applied the original TWEO formulation directly to vision classification fine-tuning and detection joint training, optimization often collapsed. For these settings, we therefore use the rescaled objective above and tune the $\lambda(t)$ schedule for each task.

A.4 KurtReg Reproduction and Hyperparameter Sweep

Following the activation-regularization setting of Nrusimha et al. (2024) for activation quantization, we implement KurtReg by applying a kurtosis penalty to the output of every linear layer. Let $Y^{(l)} \in \mathbb{R}^{B\times T\times C_l}$ be the output of layer l. For token (b, t), we compute

$$\mu_{lbt} = \frac{1}{C_l}\sum_{j=1}^{C_l} Y^{(l)}_{btj}, \qquad m_{r,lbt} = \frac{1}{C_l}\sum_{j=1}^{C_l}\left(Y^{(l)}_{btj} - \mu_{lbt}\right)^r, \quad r \in \{2, 4\}. \tag{36}$$

With sum reduction over channels, the per-token regularizer is

$$\widetilde{\kappa}_{lbt} = C_l \frac{m_{4,lbt}}{m^2_{2,lbt} + \epsilon}. \tag{37}$$

We average this quantity over tokens, samples, and the L linear layers ($L = 48$ in our ViT implementation), then add it to the cross-entropy objective:

$$R_{\text{Kurt}} = \frac{1}{L}\sum_{l=1}^{L}\frac{1}{BT}\sum_{b=1}^{B}\sum_{t=1}^{T}\widetilde{\kappa}_{lbt}, \qquad \mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{Kurt}} R_{\text{Kurt}}. \tag{38}$$

Table 5(b) varies λ_{Kurt} on ViT-B while retaining the baseline training recipe. We use 5×10^{-6} in the main comparison because it provides a favorable trade-off between FP and quantized accuracy.

A.5 Multi-Seed ImageNet-1K Results

Table 6 evaluates the variation across three training seeds for the baseline and CD. The table covers the four standard backbones and the same four quantization settings as the main pre-training comparison. CD preserves mean FP accuracy while improving mean quantized accuracy.

A.6 Official vs. Reproduced Detection Results

For detection, our reproduced checkpoints are trained and evaluated with MMDetection 3.3.0, PyTorch 2.2.1, and CUDA 12.1. We initialize the detector from the official Swin (Liu et al., 2021)

Table 6: ImageNet-1K pre-training results across three seeds. Entries report mean top-1 accuracy (%) and sample standard deviation. Quantized results report the weight and activation bit widths as WkAk. **Bold** indicates the higher mean accuracy within each model.

Model	Method	FP	RepQ-ViT W4A4	Percentile W4A6	APHQ W3A3	APHQ W4A4
ViT-B	Baseline	81.877±0.050	76.286±0.581	74.634±0.540	73.801±0.816	79.008±0.374
	CD	81.935±0.135	**78.615**±0.199	**79.213**±0.613	**74.383**±0.804	**79.277**±0.449
ViT-S	Baseline	79.933±0.013	69.032±0.695	73.888±1.098	64.881±0.739	75.073±0.079
	CD	79.947±0.044	**70.631**±0.219	**74.859**±1.262	**66.159**±0.590	**75.269**±0.135
Swin-S	Baseline	82.913±0.161	79.615±0.114	81.665±0.187	76.976±0.174	81.612±0.230
	CD	83.089±0.152	**80.162**±0.365	**81.973**±0.171	**77.504**±0.193	**81.846**±0.211
Swin-T	Baseline	81.002±0.116	76.001±0.742	78.687±0.253	74.740±0.209	79.416±0.030
	CD	81.133±0.068	**77.112**±0.413	**79.344**±0.186	**75.207**±0.260	**79.662**±0.179

Table 7: Official versus reproduced detection results. We report $\mathrm{mAP}^{\mathrm{box}}/\mathrm{mAP}^{\mathrm{mask}}$ in percent; all quantized results use W4A4. Δ is computed as reproduced minus official.

Model	Source	FP	RepQ-ViT	APHQ	GPLQ
Swin-T 1x	Official	42.6 / 39.3	13.5 / 13.7	—	37.9 / 36.8
	Reprod.	42.6 / 39.3	31.1 / 30.2	37.9 / 35.9	39.5 / 36.8
	Δ	+0.0 / +0.0	+17.6 / +16.5	—	+1.6 / +0.0
Swin-T 3x	Official	46.0 / 41.6	36.1 / 36.0	38.9 / 38.1	40.1 / 38.9
	Reprod.	45.9 / 41.5	36.6 / 35.6	39.0 / 38.0	41.7 / 39.0
	Δ	-0.1 / -0.1	+0.5 / -0.4	+0.1 / -0.1	+1.6 / +0.1
Swin-S 3x	Official	48.5 / 43.3	42.6 / 40.0	44.1 / 41.0	43.4 / 41.3
	Reprod.	48.3 / 43.2	41.4 / 38.9	40.6 / 39.2	44.5 / 40.7
	Δ	-0.2 / -0.1	-1.2 / -1.1	-3.5 / -1.8	+1.1 / -0.6

checkpoints and use the official MMDetection training configurations. Table 7 compares published official results with our reproduced baselines on the settings used in the main experiments. The official FP, RepQ-ViT (Li et al., 2023), and GPLQ (Liang et al., 2025a) numbers are taken from GPLQ paper, while the official APHQ (Wu et al., 2025) numbers are taken from APHQ paper.

The reproduced baselines do not exactly match the official numbers. Some settings are higher in our reproduction, especially Swin-T 1x under RepQ-ViT and the GPLQ results on Swin-T, while some are lower, most clearly Swin-S under RepQ-ViT and APHQ. Evaluating the official Swin-T 1x checkpoint with our RepQ-ViT pipeline gives $14.7/14.9$, close to the official $13.5/13.7$ but far below the $31.1/30.2$ obtained with our reproduced checkpoint. This check indicates that the large Swin-T 1x gap arises mainly from detector training. We suspect that these training differences come mainly from the newer software stack and different hardware generation. They do not affect the main conclusion: in the controlled comparisons in Table 2, CD consistently improves over the reproduced baseline across the detection settings and quantization algorithms.

A.7 Downstream Classification Fine-tuning

We fine-tune the official `vit_base.orig_in21k` checkpoint from timm on nine datasets: five fine-grained benchmarks (Maji et al., 2013), SUN397 (Xiao et al., 2010), EuroSAT (Helber et al., 2019) with the spatial split, SmallNORB (LeCun et al., 2004), and ImageNet-1K (Deng et al., 2009). All methods share one common fine-tuning recipe across datasets, except that ImageNet-1K uses a larger batch size and more update steps. Across methods, we also keep the same base hyperparameters, aside from the additional method-specific coefficients required by CD and TWEO (Liang et al., 2025b). Because the fine-tuning schedule is shorter and the learning rate is smaller than in pre-training, we use a larger CD strength than in pre-training. For TWEO, we use the stabilized loss rescaling the same as COCO detection. Additional fine-tuning details are provided in Appendix A.9.

Table 8: Downstream fine-tuning accuracy (%). FP accuracies are means over three seeds; W4A4 accuracies additionally report standard deviations. W4A4 results use RepQ-ViT. Drop is FP minus W4A4 accuracy, in percentage points; lower is better. Mean averages all nine datasets. Max. Act. is the maximum across module and block outputs on ImageNet-1K.

	Baseline			CD			TWEO		
Dataset	FP	W4A4	Drop	FP	W4A4	Drop	FP	W4A4	Drop
Flowers102	**98.76**	98.17±0.20	0.59	**98.76**	98.30±0.20	0.46	98.63	**98.37**±0.25	**0.26**
Pets	**95.38**	**93.98**±0.16	1.40	95.20	**93.98**±0.08	1.22	94.88	93.84±0.42	**1.04**
Aircraft	72.76	68.48±0.95	4.28	**73.30**	**70.74**±0.17	**2.56**	72.14	69.55±0.48	2.59
Cars	90.52	87.69±0.07	2.83	**90.70**	**88.99**±0.24	1.71	89.89	88.58±0.07	**1.30**
Food101	**86.68**	84.10±0.29	2.59	86.56	**84.76**±0.15	**1.81**	85.84	83.52±0.55	2.33
SUN397	68.62	63.71±0.25	4.91	**68.83**	**66.31**±0.26	**2.52**	67.15	64.30±0.41	2.85
EuroSAT	97.53	96.85±0.08	0.68	97.49	96.73±0.46	0.75	**97.54**	**96.99**±0.23	**0.54**
SmallNORB	69.01	56.72±0.76	12.29	69.01	**58.95**±0.91	**10.06**	**70.56**	57.53±0.52	13.03
ImageNet-1K	**82.98**	78.55±0.04	4.43	82.81	**79.42**±0.06	**3.39**	82.58	79.04±0.08	3.54
Mean	84.69	80.91	3.78	**84.74**	**82.02**	**2.72**	84.36	81.30	3.06
Max. Act.		1680.77			551.55			71.36	

Across these nine fine-tuning tasks, CD gives the highest mean FP accuracy, the highest mean quantized accuracy, and the smallest mean drop. Other methods remain stronger on individual datasets, especially on some FP metrics, while CD provides the best overall trade-off across the full benchmark suite. This again supports the view that controlling activations by reducing the harmful structural amplification is more effective than pushing them as low as possible.

A.8 LLM Pre-training Details

ModernLM architecture. ModernLM is our modification of nanochat (Karpathy, 2025), starting from commit `f5a0ea4`. It has 26 decoder blocks, width 1664, and 13 attention heads with head dimension 128. The number of key-value heads is also 13, so this configuration uses dense multi-head attention rather than grouped-query attention. Each block contains a SwiGLU MLP (Shazeer, 2020) with hidden width 4608. The model uses learned affine RMSNorm (Zhang & Sennrich, 2019) at the embedding, pre-attention, pre-MLP, and final normalization sites, together with per-head QKNorm (Henry et al., 2020). QKNorm is applied before RoPE (Su et al., 2024), whose base is 10,000. The input and output embeddings are untied, and the model has no linear biases or logit softcap. With a vocabulary of 32,768, the model contains exactly 995,195,136 trainable parameters.

Training. We train all models from scratch on subsets of FineWeb-Edu (Penedo et al., 2024) using AdamW and a global batch size of 524,288 tokens. GPT-2 (Radford et al., 2019) uses the GPT-2 tokenizer and a context length of 1024, whereas ModernLM uses nanochat's 32K RustBPE tokenizer and a context length of 2048. Baseline and CD runs at a given model size and token budget use the same data order and training recipe. The base weight-decay coefficient is 0.1. We set the CD coefficient to one fifth of the weight-decay coefficient, giving $\lambda_{\mathrm{cd}} = 0.02$, and reallocate the regularization budget as described in Section 3. We select checkpoints by validation BPB.

GPT-2 uses a linear warmup of 2,000 optimizer steps followed by cosine decay to one tenth of the peak learning rate. At the 30B-token scale, the peak learning rates are 3.0×10^{-4} for GPT-2 Medium and 2.5×10^{-4} for GPT-2 Large. ModernLM uses a warmup-stable-decay schedule with peak learning rate 4.0×10^{-4}: 2% of the steps are used for linear warmup, the learning rate is held constant for the next 58%, and the final 40% linearly decays to zero. We search these peak learning rates at the 30B-token scale. When increasing the budget to 60B tokens, we follow the token-scaling relation studied by the Power Scheduler (Shen et al., 2024), $\eta_{60\mathrm{B}} = \eta_{30\mathrm{B}}/\sqrt{2}$. The resulting 60B peak learning rates are approximately 1.8×10^{-4} for GPT-2 Large and 2.8×10^{-4} for ModernLM.

Evaluation and quantization. We report BPB because the GPT-2 and ModernLM tokenizers use different vocabularies and segmentations. Let $\bar{\ell}$ be the mean token-level negative log-likelihood, N_{tgt}

the number of target tokens, and N_{byte} the number of bytes represented by those targets. We compute

$$\mathrm{BPB} \simeq \frac{\bar{\ell} N_{\mathrm{tgt}}}{\ln(2) N_{\mathrm{byte}}}, \tag{39}$$

where document-boundary special tokens contribute zero bytes. The fixed GPT-2 and ModernLM validation caches contain 1,638,400 and 3,276,800 targets, respectively. Their corresponding loss-to-BPB conversion factors are 0.311810904 and 0.312266085.

We first evaluate each checkpoint in BF16. All quantization results use fake quantization to measure accuracy. Decoder-block linear layers are quantized, while the embedding layers and language-model head remain in high precision. Calibration-based methods use 128 sequences sampled from the FineWeb-Edu training split with calibration batch size 1; RTN does not require calibration. Evaluation uses 200 validation batches of size 8, with sequence lengths 1024 for GPT-2 and 2048 for ModernLM.

RTN and GPTQ (Frantar et al., 2022) use symmetric per-output-channel W4 weights. GPTQ uses Hessian dampening 0.01. AWQ (Lin et al., 2024) uses asymmetric W4 weights, activation-aware scaling, and MSE clipping; its weight-group size is 128 for GPT-2 and 512 for ModernLM. SmoothQuant (Xiao et al., 2023) uses $\alpha = 0.5$, per-channel W8 weights, and dynamic per-token A8 linear inputs, while keeping attention batch-matrix-multiplication inputs in floating point. For ModernLM, its reparameterization also covers the SwiGLU down-projection path. QuaRot (Ashkboos et al., 2024) uses W4A6KV6. Its decoder weights are quantized by GPTQ with symmetric W4 quantization, dampening 0.01. Linear inputs use dynamic symmetric per-token A6 quantization with clipping ratio 0.9. The rotated keys and values at the attention boundary use asymmetric group-128 KV6 quantization with clipping ratio 0.95, while queries remain in high precision.

A.9 Vision Task Details

Training and evaluation protocol. Training used PyTorch 2.2.1 with CUDA 12.1, FP16 AMP, FP32 parameters, and TF32 enabled. ImageNet-1K pre-training and COCO detection used distributed data parallel training with up to eight GPUs. Downstream fine-tuning ran on a single GPU, and the efficiency benchmark used two GPUs. We adapt the official open-source implementations of RepQ-ViT, APHQ, and GPLQ and preserve their core algorithmic logic and default hyperparameters, including the number of calibration images, unless otherwise noted. We add the compatibility code needed for model variant (i.e. ViT-B with AttnBias). The Percentile baseline uses the RepQ-ViT codebase with scale reparameterization and the `LogSqrt2` quantizer disabled. All vision quantization configurations use per-channel weight quantization and per-tensor activation quantization.

Efficiency benchmark. For the batch-size comparison, each method ran for five epochs with 32 data-loading workers per GPU. We averaged epochs 2–5 over three repetitions, placing all methods on the same GPU and rotating their order. The model-size comparison used fixed GPU inputs, 15 warmup steps, and 100 measured steps, yielding six paired samples across two GPU pairs. We measured peak allocated memory in separate processes for each setting after 10 warmup steps.

Downstream fine-tuning recipe. We fine-tune an ImageNet-21K-pretrained ViT-B/16 checkpoint from timm (`vit_base_patch16_224.orig_in21k`) using AdamW and AMP, with weight decay 0.05, layer-wise learning-rate decay 0.9, drop-path rate 0.1, and label smoothing 0.1. For the eight non-ImageNet datasets in Table 8, we use batch size 64, peak learning rate 5×10^{-4}, and a cosine schedule with 5 warmup epochs followed by 75 cosine epochs. Augmentation follows our timm defaults, including RandAugment and random erasing. ImageNet-1K uses four GPUs with batch size 128 per GPU, peak learning rate 10^{-3}, 500 warmup steps, and 20k total steps.

Method-specific details of detection and fine-tuning. Both downstream fine-tuning and COCO detection followed the regularization-budget rule and used a similar and simpler rescaling, normalizing W_1 to $W_1/(\|W_1\|_F/\sqrt{d_{\mathrm{out}}(W_1)})$ without matrix alignment to WD. TWEO used $\lambda(t)$ schedules tuned per task after rescaling to the task-loss magnitude.

CD-loss ablation. We detached each former matrix, so CD-loss gradients updated only latter matrices like CD. After summing the pairwise losses, we matched the total to the task-loss magnitude with a detached ratio and swept its coefficient to compare multiple regularization strengths. Since loss-based CD cannot use WD-aligned scaling, we instead normalized W_1 as above.