BREAKING THE EVEN SPLIT: CODE ALLOCATION FOR 4-BIT POST-GELU QUANTIZATION IN TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Static 4-bit post-training quantization leaves each activation tensor only sixteen codes, making their allocation a first-order design choice. In transformer MLPs, GELU confines negative outputs to a narrow, bounded interval, so nearly all of the output range lies on the long, unbounded positive side. Two-branch quantizers fit one range per sign yet split the codes evenly ($8/8$), spending half of them on the narrow negative interval. Prior methods address this asymmetry but change the code allocation jointly with the grid, region boundaries, or scale search, leaving it without a principled rule or an isolated measurement. We treat code allocation as a design variable and ask whether it can be predicted from unlabeled calibration statistics alone. Minimizing the two branches' uniform-quantization error yields a closed-form rule that sets the allocation from each sign's calibrated mass and range, without labels, gradients, Hessians, or search. Across 84 post-GELU sites, the rule tracks the reconstruction-optimal allocation (Spearman 0.95, always within one code) and leave-one-setting-out aggregation recovers a model-wide 4/12 allocation (four negative, twelve nonnegative codes) in all seven settings, the split prior work fixed by design. Held fixed across 19 vision settings (five model families, three resolutions, two datasets), $4/12$ improves top-1 accuracy over the even split in 17, by 1.09 points on average at identical BitOps, and outperforms promoting three sites to 8 bits at 8% higher cost. Without retuning, it lowers WikiText-2 perplexity versus single-range and even-split quantizers on all six language models tested when only post-GELU activations are reduced to 4 bits. The same allocation at two-sided attention inputs degrades performance in both domains, tying the gain to GELU's bounded negative lobe. Code allocation can thus be read off calibration statistics, and changing it costs no additional BitOps.

1 INTRODUCTION

Static post-training quantization (PTQ) compresses a trained network without retraining and fixes every activation range at calibration time, so inference needs no per-input scale computation. At four bits, this efficiency comes with a hard budget: each activation tensor has only sixteen codes, and a poor match between those codes and the activation distribution can erase most of a model's accuracy.

The GELU output, which feeds the second linear layer (`fc2`) of every transformer MLP, is a clear example. GELU is bounded below by -0.169971, so all negative outputs lie in an interval narrower than 0.17, whereas positive outputs are unbounded. In a ViT-B calibration sample (Figure 1(a)), the negative interval is seventeen times narrower than the fitted positive range, yet it contains most of the values. A single static range cannot serve both sides: in our setting it leaves ViT-B/224 at 0.30% ImageNet-1k top-1 (Table 1). A two-branch quantizer, which fits one range to each sign as in PTQ4ViT's twin-uniform design (Yuan et al., 2022), restores 65.32% at the same bit-width but raises a question of its own: how should the sixteen codes be divided between the two branches? The negative lobe pulls both ways—its narrow width argues for few codes, its high mass for more—so the answer is not evident by inspection.

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.

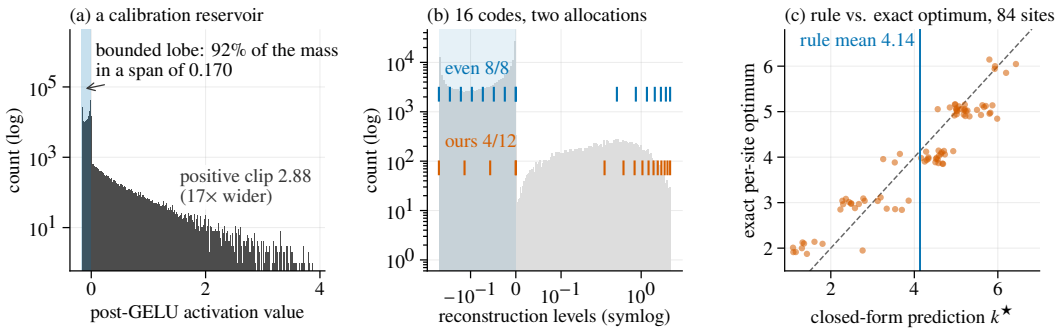

Figure 1: Why the even split is inefficient and how we choose a replacement. (a) Post-GELU activations from ViT-B: negative outputs occupy a narrow bounded interval, while positive outputs span a much wider range. (b) The sixteen codewords of the even $8/8$ split and the fixed $4/12$ allocation, which use the same four bits and BitOps; $4/12$ moves four codewords from the negative to the positive interval. (c) The calibration rule versus the reconstruction-MSE-optimal allocation among $k = 2, \ldots, 8$ at 84 post-GELU sites: after rounding, it agrees exactly at 58% of sites and is within one code everywhere (Spearman 0.95), using no labels or accuracy measurements.

Prior work answers this question within larger designs. PTQ4ViT (Yuan et al., 2022) uses the even $8/8$ split, and AMP-ViT (Tai & Wu, 2025) fixes four negative and twelve nonnegative codes by design inside a three-region quantizer with searched scales. Other methods move the activation onto shifted non-uniform grids or change the region boundaries and scale search (Jiang et al., 2025; Yvinec et al., 2023; Wu et al., 2024; Tai et al., 2023; Geng et al., 2024). Each changes the code allocation together with other parts of the quantizer, so the allocation's own contribution is unmeasured; and where an allocation exists, it is fixed by convention or heuristic, so no rule predicts it from the activation distribution.

Our central question is whether this allocation can be predicted from unlabeled calibration statistics, without validation accuracy, search, gradients, or Hessians. We depart from prior work in three ways. First, we derive the allocation in closed form from each sign's calibrated mass and range. Second, we isolate it: only the number of codes per sign changes, so every allocation has the same storage and analytic BitOps, and the even split is recovered bit-for-bit. Third, we test one predicted allocation across architectures, resolutions, datasets, and language models it was never fitted to. The rule thereby replaces design heuristics with a prediction tied to a measurable property of the activation rather than to a particular model.

The rule follows from a standard approximation. With k of the $K = 2^b$ codes on the negative range $[-S_{\text{neg}}, 0]$ and $K - k$ on the nonnegative range $[0, S_{\text{pos}}]$, each branch contributes error proportional to its probability mass times its squared step size. Minimizing the sum over k gives a closed form in which the negative share of codes grows with the cube root of $\pi_{\text{neg}} S_{\text{neg}}^2 / (\pi_{\text{pos}} S_{\text{pos}}^2)$: a branch earns more codes when it holds more mass or spans a wider range. Averaging the per-site predictions gives one model-wide allocation, $4/12$ for nearly all vision settings we examine, which we check against each site's reconstruction optimum (Figure 1(c)) and against end-to-end accuracy over $k = 2, \ldots, 8$ (Figure 2).

We claim neither GELU's lower bound (Yvinec et al., 2023), nor the $4/12$ split itself, which AMP-ViT fixes by design, nor state-of-the-art end-to-end W4A4 accuracy, since we keep several operators in floating point that the pipelines in Table 5 quantize. Our contribution is the rule that predicts this split. Because the rule followed our initial observations on vision models, its vision evidence is retrospective; leave-one-setting-out aggregation and the language models provide out-of-sample tests. Our main findings are:

1. **A closed-form rule predicts the allocation from unlabeled calibration.** Across 84 post-GELU sites from seven vision model–resolution settings, its prediction has Spearman correlation 0.95 with the reconstruction-optimal allocation and is never more than one code away, whereas a constant $4/12$ split matches only 27% of site optima at over four times

Table 1: Top-1 accuracy (%) for static 4-bit vision-transformer quantization (higher is better; full validation sets). "W4A4" uses one activation range at each post-GELU site; the even $8/8$ split and our $4/12$ allocation divide the same sixteen codes between negative and nonnegative values at identical BitOps. "+ LN fold" adds an offline RepQ-ViT-style LayerNorm reparameterization and a static W8A8 classifier head; the RepQ-ViT column reproduces the full pipeline, which quantizes more operators. Baselines use their best measured calibrator; $4/12$ is fixed for every model. Bold marks the best quantized result per row; †/‡ mark a significant win/loss for $4/12$ against the matching $8/8$ entry (paired two-sided exact McNemar, $p < .05$). BEiT-B's functional `qkv` projections remain floating point except under full RepQ-ViT.

			Baselines			Even split ($8/8$)		Ours ($4/12$)	
Dataset	Model	FP32	W4A4	SmoothQuant	RepQ-ViT	plain	+ LN fold	plain	+ LN fold
ImageNet-1k 224 px	ViT-S	81.39	25.39	0.50	**65.14**	53.23	60.94	56.02†	63.30†
	ViT-B	84.54	0.30	0.58	67.81	65.32	69.77	67.71†	**71.19**†
	ViT-L	85.84	5.55	3.83	81.40	82.95	83.01	**83.21**†	82.90
	DeiT-T	72.16	45.87	51.38	57.86	50.15	56.70	52.43†	**58.16**†
	DeiT-S	79.85	4.75	14.66	69.29	45.02	68.96	48.56†	**70.16**†
	DeiT-B	81.98	7.46	30.94	75.71	73.38	**76.63**	74.58†	76.62
ImageNet-1k 384 px	ViT-S	83.80	33.61	4.79	59.58	60.95	66.01	63.70†	**67.53**†
	ViT-B	85.99	0.62	4.31	45.40	65.60	69.97	68.30†	**71.30**†
	ViT-L	87.09	9.27	1.78	81.15	84.39	84.03	**84.68**†	84.03
	DeiT-B	83.10	24.73	61.51	76.06	79.96	79.85	**80.04**	79.76
	DeiT3-S	83.44	14.93	2.37	37.48	61.09	69.95	61.82†	**70.46**†
	DeiT3-B	85.09	12.69	65.05	75.50	81.01	80.21	**81.35**†	80.44†
BDD100K 512/518 px	ViT-S	86.72	83.16	85.21	80.65	84.65	85.34	84.87	**85.55**
	ViT-B	86.69	82.90	85.46	83.08	86.07	86.09	86.01	**86.10**
	DeiT-T	86.68	85.63	85.54	82.60	85.54	85.50	**85.97**†	85.60
	DeiT-S	86.85	76.52	83.00	80.21	83.32	84.90	84.04†	**85.22**†
	DeiT-B	86.64	70.17	84.91	85.06	85.46	85.73	85.68	**85.88**
	BEiT-B	86.74	68.23	84.46	83.07	86.24	86.45	86.29	**86.46**
	DINOv2-B	85.83	84.20	84.39	64.36	83.96	85.03	83.65‡	**85.06**

the reconstruction regret. Leave-one-setting-out aggregation recovers $4/12$ in all seven settings, on average 0.21 top-1 points below the best measured allocation (Table 8).

2. **The predicted allocation improves on the even split at identical cost.** Across 19 settings (five model families, three resolutions, two datasets), it improves top-1 over the even split in 17, by 1.09 points on average, and over the strongest same-cost baseline in 16, by 0.97 points (Table 1); all exceptions are on BDD100K. Without retuning, it also lowers WikiText-2 perplexity relative to single-range and even-split quantizers on all six language models tested when only post-GELU activations are reduced to 4 bits over a W8A8 base (Table 2).
3. **The gain is specific to the bounded lobe and cheaper than adding bits.** The same allocation at two-sided attention inputs degrades performance in both domains. Adding 8-bit activations at three post-GELU sites on top of $4/12$ costs 8% more BitOps yet lowers top-1 by 1.90 points on average, and a separate allocation per site adds no consistent gain (Table 4). The gain also persists, at a smaller margin, after a RepQ-ViT-style LayerNorm reparameterization (Li et al., 2023).

2 Related Work

Post-GELU quantization. PTQ4ViT (Yuan et al., 2022) identified post-GELU and post-SoftMax tensors as difficult and introduced twin-uniform quantization, a two-branch quantizer with the even $8/8$ split. Later designs use magnitude-based scales (Tai et al., 2023), shifted logarithmic (Jiang et al., 2025; Wu et al., 2024) or power-law (Yvinec et al., 2023) grids, adaptively merged regions (Geng et al., 2024), or hybrid logarithmic–uniform regions for heavy-tailed post-GELU activations (Zhang et al., 2025). AMP-ViT (Tai & Wu, 2025) fixes a $4/12$ allocation by design in a three-region

quantizer whose scales and region boundary are chosen with a Hessian-guided metric. Each method changes the allocation together with the grid, region boundaries, shift, or scale search; we instead vary the allocation alone and predict it from calibration statistics.

Reparameterization and reconstruction. RepQ-ViT (Li et al., 2023) folds channel-wise post-LayerNorm variation into adjacent parameters, SmoothQuant (Xiao et al., 2023) migrates activation variation into weights, and IGQ-ViT (Moon et al., 2024) adapts grouping granularity. Reconstruction methods optimize a quantized block or its weights: ERQ (Zhong et al., 2024) reduces activation and weight error in two stages, APHQ-ViT (Wu et al., 2025b) replaces GELU with ReLU and reconstructs the MLP, and FIMA-Q (Wu et al., 2025a) and LS-ViT (Hwang et al., 2026) refine the reconstruction objective. Reconstruction rebuilds more of the pipeline and can reach higher absolute W4A4 accuracy. Our rule is orthogonal to both and composes with reparameterization (Table 1).

Mixed precision. General mixed-precision methods assign bit-widths by hardware cost, curvature, or learned sensitivity (Wang et al., 2019; Dong et al., 2019; Cai & Vasconcelos, 2020); ViT-specific schemes use type-aware Fisher information (Kim et al., 2026) or layer importance and quantization sensitivity (Ranjan & Savakis, 2026); and QUIK (Ashkboos et al., 2024) keeps language-model down-projection weights and activations at higher precision. These methods spend additional bits at difficult operators; we compare against such promotion directly (Tables 3 and 4).

3 METHOD

Our goal is to determine how a two-branch quantizer should allocate a fixed code budget at post-GELU activations. The quantized tensor is the GELU output, which is the input to the second MLP Linear layer (`fc2`); we modify it in every block and keep the network, bit-width, operator scope, and calibration objective unchanged. For exact GELU, $\min_x \mathrm{GELU}(x) = -0.169971$, so the negative output spans at most 0.17 while the positive output is unbounded; approximate GELU implementations use their observed calibrated minimum instead. The even split gives both intervals the same number of codes; we keep the branches and total budget but allow their code counts to differ. Below we define the allocation, derive a calibration-only rule for choosing it, and give the procedure that deploys one model-wide allocation.

3.1 UNEQUAL CODE ALLOCATION

Let $K = 2^b$ be the number of codes at b bits. For one post-GELU site, let S_{neg} be the magnitude of the calibrated negative minimum and let S_{pos} be the fitted positive clipping point. We assign k codes to $[-S_{\mathrm{neg}}, 0]$ and $K - k$ codes to $[0, S_{\mathrm{pos}}]$. The endpoint-inclusive uniform steps are

$$\Delta_{\mathrm{neg}} = \frac{S_{\mathrm{neg}}}{k-1}, \qquad \Delta_{\mathrm{pos}} = \frac{S_{\mathrm{pos}}}{K-k-1}.$$

Values outside the calibrated support are clipped and values inside each branch are rounded to the nearest level. Assigning integer codes $0, \ldots, k-1$ to the negative branch and $k, \ldots, K-1$ to the nonnegative branch gives the reconstruction

$$\hat{x}(q) = \begin{cases} -S_{\mathrm{neg}} + q\,\Delta_{\mathrm{neg}}, & q < k, \\ (q-k)\,\Delta_{\mathrm{pos}}, & q \geq k. \end{cases}$$

Both branches include zero, so the K codewords represent $K-1$ distinct values (15 at four bits); the comparison $q < k$ identifies the branch, so no sign metadata is stored. Changing k changes neither activation storage nor analytic BitOps and uses the same two branch scales as the even split.

When $k = K/2$, the quantizer and its fitted ranges reproduce the allocation-matched even quantizer bit-for-bit, which we verify in a self-test before evaluating any model. At four bits we evaluate $k = 2, \ldots, 8$: $k = 1$ leaves the endpoint-inclusive step undefined, and $k > 8$ would give the bounded lobe more codes than the nonnegative branch; values up to $k = 14$ are valid but outside this tested range.

3.2 CHOOSING THE ALLOCATION FROM CALIBRATION

Let π_{neg} and $\pi_{\text{pos}} = 1 - \pi_{\text{neg}}$ denote the calibrated probability mass below and above zero. Using the standard approximation that a uniform step Δ contributes $\Delta^2/12$ mean squared error, the two branches have approximate distortion

$$D(k) \;\simeq\; \frac{\pi_{\text{neg}} S_{\text{neg}}^2}{12\,(k-1)^2} \;+\; \frac{\pi_{\text{pos}} S_{\text{pos}}^2}{12\,(K-k-1)^2}.$$

Since a branch with n levels has $n-1$ intervals, setting $a = k-1$ and $c = K-k-1$ (so $a + c = K-2$) and minimizing gives

$$k^\star = 1 + (K-2)\frac{\rho}{1+\rho}, \qquad \rho = \left(\frac{\pi_{\text{neg}} S_{\text{neg}}^2}{\pi_{\text{pos}} S_{\text{pos}}^2}\right)^{1/3}.$$

At four bits, $k^\star = 1 + 14\rho/(1+\rho)$. This is an approximation at few levels, not a guarantee about end-to-end accuracy; we test its local predictions against the deployed quantizer below.

The rule uses three calibration statistics: π_{neg} and S_{neg} from the reservoir, and S_{pos} taken as the even split's fitted clip (a 40-candidate search minimizing reconstruction MSE over nonnegative values), so the allocation sweep cannot influence the prediction. Only S_{neg} is bounded by GELU; the mass and positive clip remain data- and model-dependent. Replacing the observed S_{neg} by 0.169971 changes our vision predictions by under 0.01 codes.

Example. At block 0 of ViT-B/224, the calibration reservoir gives $\pi_{\text{neg}} = 0.925$, $S_{\text{neg}} = 0.16997$, and $S_{\text{pos}} = 3.40$. Thus $\rho = (0.925 \cdot 0.170^2/(0.075 \cdot 3.40^2))^{1/3} = 0.313$ and $k^\star = 4.34$, which rounds to four negative and twelve nonnegative codes. Directly minimizing reconstruction MSE for this site's deployed quantizer also selects four.

3.3 CALIBRATION AND DEPLOYMENT PROCEDURE

Given unlabeled calibration data:

1. At every post-GELU site, collect a reservoir and compute π_{neg}, S_{neg}, and the even-split positive clip S_{pos}.
2. Evaluate $k^\star$ at each site, take the unweighted mean across the model's sites, round it, and restrict it to the tested range $2, \ldots, K/2$.
3. With this model-wide k fixed, refit each site's positive clip for the $K-k$ positive levels actually deployed; the negative range remains the observed bounded support.
4. Apply the same k to every post-GELU site at inference.

No labels, gradients, Hessians, validation accuracy, or search over k are used; the 40-candidate searches in Steps 1 and 3 fit ranges, exactly as for the even split.

Figure 1(c) evaluates the rule on 84 post-GELU sites from seven vision model–resolution settings with calibration reservoirs and complete allocation curves: after rounding, it matches the reconstruction-MSE optimum at 58% of sites, is always within one code (Spearman 0.95), and has mean reconstruction regret 1.68% versus 7.17% for a fixed $k = 4$. These are local reconstruction results, not accuracy claims. Averaged per setting, the rule selects $4/12$ in six of seven settings and $5/11$ for DeiT-T/224; leave-one-setting-out aggregation selects $4/12$ every time, within 0.01 top-1 points of fixed $4/12$. We therefore use $4/12$ in every headline setting without validation-based selection. Because the rule followed our initial allocation observation, this is retrospective support, not a preregistered prediction.

4 EXPERIMENTAL SETUP

Evaluation matrix. We evaluate 19 vision-transformer settings in three arms. A high-resolution ImageNet-1k (Deng et al., 2009) arm evaluates six checkpoints at their native 384-pixel resolution,

including the 24-block ViT-L; a six-model 224-pixel arm matches the resolution of most published W4A4 ViT comparisons; and a driving arm contains seven ViT, DeiT, BEiT, and DINOv2 models fine-tuned for BDD100K weather classification (Yu et al., 2020), evaluated at 512^2 (518^2 for DINOv2) on native 1280×720 images without upsampling. Checkpoints use `timm` (Wightman, 2019) implementations of ViT (Dosovitskiy et al., 2021), DeiT (Touvron et al., 2021), DeiT III (Touvron et al., 2022), BEiT (Bao et al., 2022), and DINOv2 (Oquab et al., 2024).

We report top-1 accuracy on all 50,000 ImageNet-1k and all 8,843 BDD100K validation images. The largest BDD weather class holds 60.5% of the validation set, so small changes on that arm can be driven by the majority class; we use it mainly to test whether the ordering transfers to high-resolution driving imagery.

Quantization scope. Inside each transformer block, Linear weights are quantized to four bits per output channel and their input activations use static per-tensor four-bit ranges. The code allocation is changed only at the post-GELU inputs of `fc2`; other quantized activations use the harness's standard static quantizer. Attention matrix multiplications, the post-SoftMax map, patch embedding, and LayerNorm remain floating point, as do BEiT's functional `qkv` projections; the classifier is W8A8 (static in the LayerNorm-folded rows). Accuracies use fake quantization and compute is reported as analytic BitOps, so absolute numbers are not directly comparable to pipelines that quantize the operators we keep in floating point.

Calibration and comparison protocol. Static ranges are calibrated from 128 unlabeled training images (seed 42); the headline $8/8$ versus $4/12$ comparison is repeated with seeds 43 and 44 on ImageNet-1k/224, with all draws in the appendix. Each baseline in Table 1 reports the best of its predefined calibrators: four for a single range, two for SmoothQuant, two for the even split, and plain and bit-appropriately recalibrated variants for A8 promotion. Our row uses the same fixed $4/12$ allocation and one calibration procedure for every model, so the protocol favors the baselines while avoiding validation-based selection for our method.

The $8/8$ reference matches PTQ4ViT's code allocation but not its Hessian-guided range search or broader quantization scope, so we call it the allocation-matched even reference rather than PTQ4ViT. Table 1 reports a full RepQ-ViT reproduction separately from the two columns that use only its offline LayerNorm reparameterization.

Statistical tests. Pairwise verdicts use stored per-image correctness vectors and a two-sided exact McNemar test ($p < .05$, unadjusted for multiple comparisons). Configurations from the same job share the calibration draw, image order, and evaluation code; comparisons assembled across jobs are accepted only when FP32 accuracy, FP32 per-image correctness, and labels agree exactly, which holds for all 19 settings.

5 Results

5.1 The predicted $4/12$ allocation improves on the even split at the same cost

Table 1 is the primary comparison. In the plain pipeline, changing only the post-GELU code allocation from the even $8/8$ split to $4/12$ improves top-1 accuracy in 17 of 19 settings, by 1.09 points on average, with 13 gains significant under paired McNemar tests. Relative to the strongest same-cost baseline for each model (single range, SmoothQuant, or $8/8$), $4/12$ improves 16 of 19 settings by 0.97 points on average. The largest gain over $8/8$ is 3.54 points on DeiT-S/224 ($45.02\% \rightarrow 48.56\%$). The single-range and SmoothQuant columns also show that channel migration alone does not repair post-GELU failures in this harness (e.g., 0.58% for SmoothQuant on ViT-B/224), which motivates the two-branch quantizer; our question is how its sixteen codes are allocated.

The effect varies across arms: the mean gain over $8/8$ is 2.08 points on ImageNet-1k/224, 1.15 on ImageNet-1k/384, and 0.18 on BDD100K. The 224-pixel arm is a conventional reference for published W4A4 ViT results; the native 384-pixel and driving arms define the paper's high-resolution scope. On BDD, the strongest baselines already lie close to FP32, leaving less room for gains. Across calibration draws in the 224-pixel arm, $4/12$ wins all 18 model–draw comparisons (seeds 42–44).

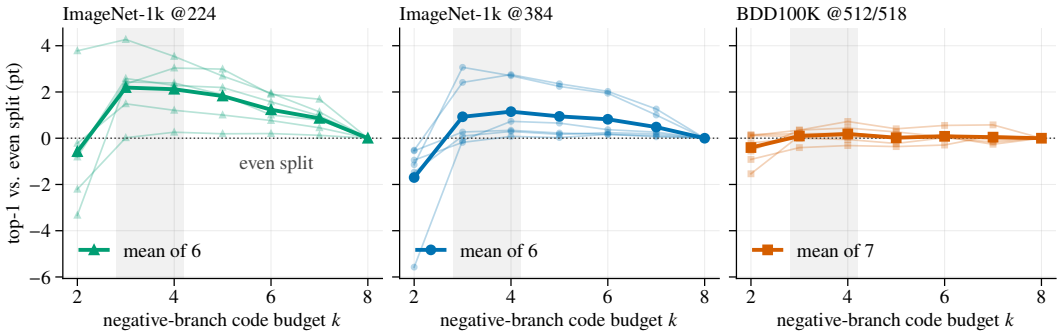

Figure 2: Accuracy across code allocations. Each curve reports top-1 accuracy relative to the even $8/8$ split as the number of negative-branch codes k varies; faint lines are individual models and bold lines are arm means. All points use sixteen codes and the same BitOps. The shaded $k = 3$–4 region contains the best value for 15 of 19 settings and is the region selected by the calibration rule.

The gain is not universal. Relative to $8/8$, BDD ViT-B is a tie (-0.06 points, $p = .70$) and BDD DINOv2-B a significant loss ($83.96\% \rightarrow 83.65\%$, $p = .029$); relative to the strongest same-cost baseline, BDD ViT-S is an additional non-significant loss. The evidence thus supports an average improvement, not a guarantee for every architecture.

5.2 The useful allocation is a broad interior region

Figure 2 sweeps $k = 2, \ldots, 8$, from a strongly positive-weighted allocation to the even split. Accuracy peaks in an interior region around $k = 3$–4 and usually declines toward $k = 8$: the best value is 3 or 4 in 15 of 19 settings, and at least three values lie within half a point of the best in 15 of 19. On BDD the curve moves by less than one point, so this arm says little about the choice of k. The calibration rule selects the same region without validation accuracy (Figure 1(c)); leave-one-setting-out aggregation gives $4/12$ in all seven settings where it was evaluated, so every headline row uses this fixed allocation.

5.3 Cross-domain check on language models

The bounded GELU lobe is not specific to images. We test six pretrained language models from the Pythia (Biderman et al., 2023), BLOOM (BigScience Workshop, 2022), Phi-2 (Javaheripi & Bubeck, 2023), and GPT-2 (Radford et al., 2019) families (124M–2.7B parameters) on WikiText-2 (Merity et al., 2017). With the allocation fixed on vision, $4/12$ gives lower perplexity than both a single range and the even split on all six models (Table 2). As a matched control, applying $4/12$ instead of $8/8$ at the two-sided attention inputs worsens perplexity on all six, so the benefit is specific to the bounded GELU output rather than to unequal allocation in general. In the appendix, the rule, the pooled reconstruction optimum over 84 Pythia/BLOOM sites, and all six perplexity optima fall in the $k = 3$–4 region, supporting region-level rather than exact transfer; an all-Linear W4A4 stress test, which remains severely degraded even with SmoothQuant-style channel migration, is reported there as well.

5.4 Three-site A8 promotion does not beat code reallocation

Table 3 compares $4/12$ with single-range A4 pipelines that promote three or all post-GELU sites to A8. Promoting three sites costs 8% more BitOps yet gives a larger mean accuracy drop on every arm, driven at 224 pixels by ViT-L falling to 9.60%. Promoting all sites costs 33% more and is worse than $4/12$ at 384 pixels, the same on BDD after rounding, and 0.2 points better at 224 pixels. Uniform W8A8 has the smallest drop but four times the BitOps of uniform W4A4. Because these rows also change the base quantizer, Table 4 adds the same three A8 sites on top of $4/12$; this matched variant also loses.

Relative to the even split, reallocating codes adds no analytic compute or storage cost: both quantizers store four bits per activation, use two branch scales, and have identical BitOps and packed size.

Table 2: WikiText-2 perplexity (lower is better; compare within rows, since tokenizers differ). "Full" is unquantized; the 8-bit base quantizes transformer Linear weights and inputs to 8 bits. The last three columns additionally reduce post-GELU activations to 4 bits with one 16-code range, the even $8/8$ split, or our $4/12$ allocation, fixed on vision. ⋆GPT-2 uses 1024-token blocks, others 2048. Bold marks the best 4-bit treatment. This check is narrower than the vision W4A4 setting.

	Reference		GELU output at 4 bits		
Model	Full	8-bit base	One range	Even $8/8$	Ours $4/12$
Pythia-160M	26.65	34.81	71.07	63.34	**50.77**
Pythia-410M	15.95	19.28	60.73	67.13	**45.95**
Pythia-1.4B	11.79	23.27	52.39	36.78	**33.09**
BLOOM-560M	22.37	24.82	52.98	30.68	**28.54**
Phi-2 (2.7B)	9.71	10.95	30.09	13.94	**12.57**
GPT-2 (124M)⋆	29.95	36.04	90.58	51.99	**42.76**

Table 3: Mean top-1 drop from FP32 (points; lower is better) and analytic BitOps relative to uniform W4A4, averaged over six ImageNet-1k/224, five ImageNet-1k/384 (ViT-L excluded, as its A8 rows were not measured), and seven BDD100K models. Promotion rows use a single-range A4 base, raise three or all post-GELU sites to A8, and use their best measured calibration.

		Mean accuracy drop from FP32 (percentage points ↓)		
Treatment	Relative BitOps	IN-1k 224 px	IN-1k 384 px	BDD100K 512/518 px
$4/12$ allocation (ours)	$1.00\times$	17.2	13.2	1.4
Single + 3-site A8	$1.08\times$	37.2	22.6	2.4
Single + all-site A8	$1.33\times$	17.0	16.4	1.4
Uniform W8A8	$4.00\times$	2.7	1.7	0.0

In our eager FP32 harness, their median ViT-B/384 forward passes (batch 16, 50 repetitions) differ by 0.17% (226.91 vs. 226.52 ms); these are harness timings, not deployed INT4 latency.

5.5 The gain persists after LayerNorm reparameterization

The "+ LN fold" columns in Table 1 test whether the gain survives a stronger pipeline. The RepQ-ViT-style reparameterization folds post-LayerNorm channel variation into the following `qkv`/`fc1` weights offline; it preserves full-precision outputs and does not touch the post-GELU tensor or its quantizer. Both folded rows use a static W8A8 classifier head.

Folding changes $4/12$ accuracy by -0.91 to $+21.60$ points across settings (mean $+2.99$; positive in 14 of 19), so its benefit is highly model-dependent. Within the folded pipeline, $4/12$ still improves on $8/8$ in 15 of 19 unrounded comparisons, by 0.56 points on average; nine gains are significant and none significantly favors $8/8$. The smaller margin (0.56 versus 1.09) does not support an additive-benefit claim, but the allocation remains useful when the post-LayerNorm bottleneck is treated separately.

5.6 Matched controls

Table 4 tests whether the benefit is specific to post-GELU activations, whether each site needs its own allocation, and whether adding precision is preferable. Moving $4/12$ to the two-sided `qkv` inputs reduces top-1 by 5.90 points on average, significantly worse in 17 of 18 informative settings and better in one, so the effect is strongly site-specific though not exception-free. Choosing an allocation per post-GELU site by calibration MSE changes the mean by -0.11 points (one significant win, five losses), and adding A8 at three selected sites changes it by -1.90 points (three wins, ten losses). Neither added adaptivity nor added precision consistently beats the fixed $4/12$ allocation.

Table 4: Matched controls for the fixed $4/12$ allocation. Δ is the mean top-1 change from the stated reference (positive is better); "Sig. W/L" counts significant wins/losses (paired two-sided exact McNemar, $p < .05$). Each comparison uses rows from the same job. BEiT-B is excluded from the qkv row because its functional qkv projections remain floating point.

Treatment	Reference	n	Mean Δ (pt) $\uparrow$	Sig. W / L
$4/12$ at qkv	one range	18	-5.90	1 / 17
Per-site MSE allocation	fixed $4/12$	19	-0.11	1 / 5
$4/12$ + 3-site A8	fixed $4/12$	19	-1.90	3 / 10

Table 5: Published ImageNet-1k/224 W4A4 context (top-1 %), not a controlled leaderboard. Our $4/12$ row includes the LayerNorm fold but keeps attention matmuls, the post-SoftMax map, and patch embedding in floating point with a W8A8 classifier; prior methods quantize these at four bits. †Searched, shifted logarithmic grid. ‡Gradient-based reconstruction with GELU replaced by ReLU. FP32 and ours are measured here; other values are published.

Method	Calibration images	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B
FP32	—	81.39	84.54	72.16	79.85	81.98
Uniform grid; no retraining or architecture change						
RepQ-ViT	32	65.05	68.48	57.43	69.03	75.61
Ours $(4/12)$	128	63.30	71.19	58.16	70.16	76.62
Specialized quantizer or modified network						
AMP-ViT	32	55.88	61.84	55.62	68.43	76.14
AdaLog†	32	72.75	79.68	63.52	72.06	78.03
APHQ-ViT‡	1,024	76.07	82.41	66.66	76.40	80.21

5.7 Comparison with published four-bit methods

Table 5 gives published context at the common ImageNet-1k/224 setting. Our $4/12$ row exceeds AMP-ViT on all five models ($+0.5$–9.4 points) and RepQ-ViT on four of five, but trails AdaLog by 1.4–9.5 and APHQ-ViT by 3.6–12.8 points. Because operator scopes differ, these are absolute comparisons, not controlled ablations; our claim is narrower: within a two-branch quantizer, changing only the code allocation improves on the even split at no additional cost.

6 Discussion and Limitations

What the rule claims: The rule predicts local reconstruction-MSE optima and, after aggregation, an allocation chosen without labels or accuracy search that transfers from vision to language models (Table 2). This retrospective result supports explanation and transfer, not a prospective guarantee. Other limitations: "No search" excludes allocation search only; our method and the even split retain the same 40-candidate range calibration. Near-FP32 BDD curves bound—not drive—the aggregate gain and include a genuine DINOv2-B loss. Adaptivity covers one selector and objective, and the rule covers only GELU; better per-site selectors, other bounded-below activations, and robustness under distribution shift remain open (Araghi et al., 2026).

7 Conclusion

We derived a calibration rule that assigns fewer codes to GELU's bounded negative lobe and more to its long positive range. Its predicted $4/12$ allocation improves on the even split in 17 of 19 vision settings at identical BitOps and, without retuning, lowers perplexity on all six language models. Matched controls show the benefit is not generic: moving $4/12$ to two-sided attention inputs degrades performance, while per-site allocation adds no consistent gain. Nor is extra precision preferable: promoting three sites to A8 costs 8% more and lowers mean vision accuracy. These results make code allocation a simple, calibration-derived design choice for low-bit post-GELU quantization.

AI USE STATEMENT

The research questions, method, derivation of the allocation rule, experimental design, implementation, experiments, and interpretation of results are entirely the authors' own work. We used generative AI tools only for tasks with recommended disclosure, all related to edit and review code, modify figures, and the manuscript: editing parts of the paper to improve readability and refine its wording, drafting a small number of table captions, and help with LaTeX formatting. We reviewed all AI-assisted edits and checked that they do not change the technical content or any reported result. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The code for all experiments is provided as a zip file in the supplementary material. All of our experimental results, including those in the appendix, can be reproduced with it.

REFERENCES

Hamidreza Yaghoubi Araghi, Parastoo Pilevar, and Ming C Lin. Recti-Q: Feature-space rectification for out-of-distribution-robust quantized perception in edge robotics. *arXiv preprint arXiv:2607.18540*, 2026.

Saleh Ashkboos, Ilia Markov, Elias Frantar, Tingxuan Zhong, Xincheng Wang, Jie Ren, Torsten Hoefler, and Dan Alistarh. QUIK: Towards end-to-end 4-bit inference on generative large language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3355–3371, 2024.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *International Conference on Learning Representations (ICLR)*, 2022.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning (ICML)*, volume 202, pp. 2397–2430, 2023.

BigScience Workshop. BLOOM: A 176B-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

Zhaowei Cai and Nuno Vasconcelos. Rethinking differentiable search for mixed-precision neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2349–2358, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. HAWQ: Hessian aware quantization of neural networks with mixed-precision. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 293–302, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Xinkuang Geng, Siting Liu, Leibo Liu, Jie Han, and Honglan Jiang. QUQ: Quadruplet uniform quantization for efficient vision transformer inference. In *ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2024. doi: 10.1145/3649329.3656516.

Hyunha Hwang, Xuan Truong Nguyen, and Hyuk-Jae Lee. LS-ViT: Least-squares hessian based block reconstruction for low-bit post-training quantization of vision transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 33588–33597, 2026.

Mojan Javaheripi and Sébastien Bubeck. Phi-2: The surprising power of small language models. Microsoft Research Blog, 2023. https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/.

Yanfeng Jiang, Ning Sun, Xueshuo Xie, Fei Yang, and Tao Li. ADFQ-ViT: Activation-distribution-friendly post-training quantization for vision transformers. *Neural Networks*, 186:107289, 2025. doi: 10.1016/j.neunet.2025.107289.

Minjun Kim, Jaeri Lee, Jongjin Kim, Jeongin Yun, Yongmo Kwon, and U Kang. LampQ: Towards accurate layer-wise mixed precision quantization for vision transformers. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 5691–5699, 2026. doi: 10.1609/aaai.v40i7.37489.

Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. RepQ-ViT: Scale reparameterization for post-training quantization of vision transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17181–17190, 2023.

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

Jaehyeon Moon, Dohyung Kim, Junyong Cheon, and Bumsub Ham. Instance-aware group quantization for vision transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16132–16141, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

Navin Ranjan and Andreas Savakis. Mix-QViT: Mixed-precision vision transformer quantization driven by layer importance and quantization sensitivity. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3599–3609, 2026.

Yu-Shan Tai and An-Yeu Wu. AMP-ViT: Optimizing vision transformer efficiency with adaptive mixed-precision post-training quantization. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6828–6837, 2025.

Yu-Shan Tai, Ming-Guang Lin, and An-Yeu Wu. TSPTQ-ViT: Two-scaled post-training quantization for vision transformer. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10096817.

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 (ICML)*, volume 139, pp. 10347–10357, 2021.

Hugo Touvron, Matthieu Cord, and Hervé Jégou. DeiT III: Revenge of the ViT. In *European Conference on Computer Vision (ECCV)*, pp. 516–533, 2022.

Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. HAQ: Hardware-aware automated quantization with mixed precision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8612–8620, 2019.

Ross Wightman. PyTorch image models. https://github.com/huggingface/pytorch-image-models, 2019.

Zhuguanyu Wu, Jiaxin Chen, Hanwen Zhong, Di Huang, and Yunhong Wang. AdaLog: Post-training quantization for vision transformers with adaptive logarithm quantizer. In *European Conference on Computer Vision (ECCV)*, pp. 411–427, 2024.

Zhuguanyu Wu, Shihe Wang, Jiayi Zhang, Jiaxin Chen, and Yunhong Wang. FIMA-Q: Post-training quantization for vision transformers by fisher information matrix approximation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14891–14900, 2025a.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9686–9695, 2025b.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning (ICML)*, volume 202, pp. 38087–38099, 2023.

Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2636–2645, 2020.

Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. PTQ4ViT: Post-training quantization for vision transformers with twin uniform quantization. In *European Conference on Computer Vision (ECCV)*, pp. 191–207, 2022.

Edouard Yvinec, Arnaud Dapogny, Matthieu Cord, and Kevin Bailly. PowerQuant: Automorphism search for non-uniform quantization. In *International Conference on Learning Representations (ICLR)*, 2023.

Wenlun Zhang, Yunshan Zhong, Shimpei Ando, and Kentaro Yoshioka. AHCPTQ: Accurate and hardware-compatible post-training quantization for segment anything model. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22383–22392, 2025.

Yunshan Zhong, Jiawei Hu, You Huang, Yuxin Zhang, and Rongrong Ji. ERQ: Error reduction for post-training quantization of vision transformers. In *International Conference on Machine Learning (ICML)*, volume 235, pp. 61664–61680, 2024.

A Appendix of Breaking the Even Split: Code Allocation for 4-Bit Post-GELU Quantization in Transformers

This appendix provides supporting experimental detail, calibration-draw replicates, complete allocation curves, rule-validation results, language-model analyses, matched controls, and fake-quantization timing. It does not change the quantization scope or claims of the main text. All accuracy comparisons use stored per-example correctness vectors; no result in this appendix is estimated from a figure.

B Experimental Details

B.1 Vision evaluation

The vision evaluation contains 19 model–resolution settings. ImageNet-1k results use all 50,000 validation images. The conventional 224-pixel arm has six ViT/DeiT checkpoints, including 24-block ViT-L. The native 384-pixel arm has six ViT/DeiT/DeiT3 checkpoints and also includes ViT-L. The driving arm contains seven ViT, DeiT, BEiT, and DINOv2 models fine-tuned for BDD100K weather classification. They are evaluated on all 8,843 validation images at 512×512 pixels, except DINOv2-B at 518×518; source images are natively 1280×720, and no image is upsampled.

Inside each transformer block, Linear weights use four-bit per-output-channel quantization and their inputs use static per-tensor four-bit activation ranges. The unequal allocation is applied only to the post-GELU input of the second MLP Linear. Attention matrix multiplications, the post-SoftMax map, patch embedding, and LayerNorm remain floating point, and the classifier is W8A8. BEiT-B implements `qkv` functionally, so those projections remain floating point in our rows. Evaluation uses eager fake quantization; BitOps are analytic rather than measured kernel operations.

Static ranges are calibrated from 128 unlabeled training images. The primary tables use calibration seed 42. Baseline selection is deliberately favorable to each baseline: the paper reports its strongest predefined calibrator, while the proposed row always uses the fixed $4/12$ allocation and the same calibration procedure. The replicate study in Appendix C instead holds the even-split calibrator fixed, which isolates calibration-draw sensitivity.

Table 6 summarizes the vision evaluation matrix. Every pairwise vision verdict uses a two-sided exact McNemar test on the per-image correctness vectors. Across-job comparisons are accepted only when the FP32 accuracy, FP32 correctness vector, and label vector agree exactly. Significance uses $p < .05$ without correction for multiple comparisons.

B.2 Language-model evaluation

The text-domain check evaluates Pythia-160M, Pythia-410M, Pythia-1.4B, BLOOM-560M, Phi-2, and GPT-2 on WikiText-2. It is a narrower mechanism check: the 8-bit base quantizes transformer Linear weights and their inputs to 8 bits, then only post-GELU inputs are reduced to four-bit activations. Pythia, BLOOM, and Phi-2 use all complete 2,048-token test blocks (140, 139, and 140 blocks, respectively); GPT-2 uses 280 complete 1,024-token blocks. Because tokenizers differ, perplexities are compared only within a model. Calibration uses 16 training sequences. Reported paired block-NLL tests are descriptive screens because adjacent text blocks need not be independent.

Table 6: Vision evaluation matrix. Calibration images are unlabeled training examples; evaluation uses the complete validation split.

Arm	Models	Eval. images	Calib. images
ImageNet-1k, 224 px	6	50,000	128
ImageNet-1k, 384 px	6	50,000	128
BDD100K, 512/518 px	7	8,843	128

Table 7: Calibration-draw replication. Each seed cell reports "even $8/8$ / fixed $4/12$" top-1 accuracy; mean and range summarize the three gains, $4/12 - 8/8$. Accuracies are displayed to two decimals, whereas summaries use unrounded values when available, so recomputation from displayed cells can differ by 0.01. Dagger/double dagger: significant win/loss for $4/12$ under a paired two-sided exact McNemar test ($p < .05$).

Dataset	Model	Top-1 accuracy (%): even $8/8$ / fixed $4/12$			$4/12 - 8/8$ (points)	
		Seed 42	Seed 43	Seed 44	Mean	Range
ImageNet-1k 224	ViT-S	53.23 / 56.02†	47.95 / 51.91†	54.28 / 58.67†	+3.71	1.60
	ViT-B	65.32 / 67.71†	63.33 / 66.15†	64.02 / 66.96†	+2.72	0.56
	ViT-L	82.95 / 83.21†	82.88 / 82.99	82.75 / 82.99†	+0.20	0.15
	DeiT-T	50.15 / 52.43†	51.61 / 53.67†	49.66 / 51.95†	+2.21	0.22
	DeiT-S	45.02 / 48.56†	43.34 / 47.48†	43.14 / 47.21†	+3.92	0.59
	DeiT-B	73.38 / 74.58†	73.12 / 74.28†	72.27 / 73.78†	+1.28	0.35
ImageNet-1k 384 px	ViT-S	60.95 / 63.70†	59.32 / 62.25†	60.82 / 63.93†	+2.93	0.36
	ViT-B	65.60 / 68.30†	66.45 / 68.50†	68.47 / 70.68†	+2.32	0.65
	ViT-L	84.39 / 84.68†	84.60 / 84.86†	84.54 / 84.55	+0.19	0.28
	DeiT-B	79.96 / 80.04	80.05 / 79.97	79.90 / 79.92	+0.01	0.16
	DeiT3-S	61.09 / 61.82†	60.27 / 60.79†	61.98 / 62.68†	+0.65	0.21
	DeiT3-B	81.01 / 81.35†	81.21 / 81.45†	81.08 / 81.27	+0.26	0.15
BDD100K 512/518 px	ViT-S	84.65 / 84.87	84.67 / 84.79	84.73 / 84.78	+0.13	0.17
	ViT-B	86.07 / 86.01	85.99 / 85.93	85.97 / 85.92	−0.05	0.01
	DeiT-T	85.54 / 85.97†	85.75 / 85.65	85.75 / 85.68	+0.09	0.54
	DeiT-S	83.32 / 84.04†	83.91 / 83.39‡	83.30 / 83.06	−0.01	1.24
	DeiT-B	85.46 / 85.68	85.23 / 85.46	85.22 / 85.55†	+0.26	0.10
	BEiT-B	86.24 / 86.29	86.17 / 86.19	86.27 / 86.20	+0.00	0.12
	DINOv2-B	83.96 / 83.65‡	83.90 / 83.74	84.04 / 84.13	−0.13	0.41

C Calibration-Draw Robustness

We repeat the controlled comparison between the allocation-matched even $8/8$ split and the fixed $4/12$ allocation with calibration seeds 42, 43, and 44. These are *calibration-draw* seeds: the model weights and evaluation set do not change. Within each of the 19 settings, the three artifacts share bit-identical labels and FP32 correctness vectors. All three draws also use the same RTX A5000 GPU model. Two historical seed-42 settings produced on other GPU models were rerun on an A5000 before forming this table.

Table 7 reports every draw. Each seed column is "$8/8$ / $4/12$" top-1 accuracy. A dagger or double dagger marks a significant $4/12$ win or loss for that draw. The last two columns summarize the three paired gains in percentage points. This table fixes the even calibration procedure at `twin8_ref`; it is therefore a controlled draw study, not a replicate of the per-setting best-of-calibrators selection in Table 1.

Across ImageNet-1k/224, $4/12$ exceeds $8/8$ in all 18 model–draw comparisons (mean gain $+2.34$ points; worst draw $+0.11$). At 384 pixels it does so in 17 of 18 comparisons (mean $+1.06$; worst -0.08). The near-FP32 BDD100K arm is less stable: $4/12$ is higher in 11 of 21 comparisons (mean $+0.04$; worst -0.52). Thus the draw study confirms the 224-pixel ordering claimed in the main text while explicitly bounding its transfer to the high-resolution driving arm.

D Validation of the Allocation Rule

Table 8 evaluates the closed form as both a local reconstruction predictor and a model-level decision procedure. At 84 post-GELU sites from seven ImageNet model–resolution settings, the rounded prediction matches the tested reconstruction-MSE optimum at 58% of sites, is within one code at every site, and has mean absolute error 0.42. Its mean relative reconstruction regret is 1.68%, compared with 7.17% for the fixed constant $k = 4$ and 58.67% for the even split. These are local

Table 8: Validation of the calibration rule on ImageNet-1k. (a) Agreement of each predictor with the per-site reconstruction-MSE optimum over 84 post-GELU sites; the regret is the mean relative increase in reconstruction MSE. (b) Model-level decisions: the model-wide allocation k given by the rounded mean prediction over the model's own sites (same-model) and by leave-one-setting-out aggregation (leave-one-out), with top-1 accuracy at the same-model k, at fixed $k = 4$, and at the best measured allocation, whose k is given in parentheses.

(a) Site-level prediction against the reconstruction optimum (84 post-GELU sites)

Predictor	Exact match	Within ± 1 code	MAE (codes)	Mean MSE regret
Closed form	58%	100%	0.42	1.68%
Constant $k = 4$	27%	82%	0.90	7.17%
Even split, $k = 8$	0%	0%	3.98	58.67%

(b) One model-wide allocation per model

	Chosen allocation k		Top-1 accuracy (%) ↑		
Model	Same-model k	Leave-one-out k	At same-model k	At fixed $k = 4$	Best measured top-1 (k)
ImageNet-1k @224					
DeiT-B	4	4	74.58	74.58	74.67 (3)
DeiT-S	4	4	48.56	48.56	49.29 (3)
DeiT-T	5	4	52.35	52.43	52.74 (3)
ViT-B	4	4	67.71	67.71	67.76 (3)
ViT-S	4	4	56.02	56.02	56.02 (4)
ImageNet-1k @384					
DeiT3-S	4	4	61.82	61.82	61.82 (4)
ViT-B	4	4	68.30	68.30	68.56 (3)

reconstruction results, not end-to-end accuracy claims. The seven stored rule reservoirs do not include BDD100K.

E Full Vision Allocation Curves

Tables 9 and 10 report top-1 accuracy as k, the number of four-bit codes assigned to the bounded negative GELU lobe, varies from 2 to 8. The other $16 - k$ codes represent the nonnegative branch, and $k = 8$ is the allocation-matched even split. Eighteen settings store the full curve in one artifact or in jobs linked by a bit-identical $k = 8$ row. ViT-L/224 is assembled from four seed-42 RTX A5000 jobs only after confirming exact equality of FP32 accuracy, FP32 correctness, and labels across all four; its repeated $k = 8$ and $k = 4$ rows are also bit-identical across independent artifacts.

F Language-Model Analyses

F.1 Local reconstruction rule

Across the 84 Pythia/BLOOM post-GELU sites in Table 11, the local reconstruction optimum has median four and range two to six, while the pooled closed-form prediction is 4.40 codes and rounds to $k = 4$, inside the $k = 3$–4 region. The prediction consumes the calibrated negative mass, negative span, and positive clip. Observed negative spans reach GELU's analytic bound, which is why substituting the bound leaves the displayed predictions unchanged at two decimals. The language-model experiment transfers the vision-fixed constant $k = 4$; it does not use these statistics to select a model-specific allocation. Applied retrospectively, the per-model means in Table 11 give $k = 5$ for BLOOM-560M and Pythia-1.4B and $k = 4$ for Pythia-160M and Pythia-410M. Thus the pooled and per-model predictions are all within one integer code of the $k = 3$–4 region.

Table 9: ImageNet-1k allocation curves (top-1 %). Bold marks the best measured allocation in each row. ViT-L/224 combines four seed-42 jobs under the identity and repeated-row gates described in Appendix E.

Dataset	Model	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	Best k
ImageNet-1k 224 px	ViT-S	49.65	55.33	**56.02**	55.97	54.88	54.67	53.23	4
	ViT-B	64.51	**67.76**	67.71	67.20	66.33	66.10	65.32	3
	ViT-L	80.74	82.97	**83.21**	83.14	83.15	83.07	82.95	4
	DeiT-T	49.47	**52.74**	52.43	52.35	51.73	51.12	50.15	3
	DeiT-S	48.80	**49.29**	48.56	47.71	46.97	46.16	45.02	3
	DeiT-B	73.09	**74.67**	74.58	74.38	74.14	73.83	73.38	3
ImageNet-1k 384 px	ViT-S	60.44	63.37	**63.70**	63.31	62.98	62.22	60.95	4
	ViT-B	64.43	**68.56**	68.30	67.84	67.54	66.60	65.60	3
	ViT-L	82.92	84.49	**84.68**	84.56	84.63	84.58	84.39	4
	DeiT-B	79.02	79.78	80.04	79.99	**80.13**	80.08	79.96	6
	DeiT3-S	55.51	60.99	**61.82**	61.74	61.46	61.35	61.09	4
	DeiT3-B	80.46	81.28	**81.35**	81.23	81.18	81.07	81.01	4

Table 10: BDD100K allocation curves (top-1 %). Excluding $k = 2$, every curve spans less than 0.75 points, so this arm gives little evidence for selecting one exact allocation over another.

Dataset	Model	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	Best k
BDD100K 512/518 px	ViT-S	84.81	84.85	**84.87**	84.69	84.71	84.39	84.65	4
	ViT-B	85.86	86.07	86.01	85.84	**86.09**	85.90	86.07	6
	DeiT-T	85.64	85.88	**85.97**	85.80	85.58	85.59	85.54	4
	DeiT-S	82.77	83.67	**84.04**	83.73	83.87	83.90	83.32	4
	DeiT-B	83.92	85.56	**85.68**	85.49	85.62	85.38	85.46	4
	BEiT-B	**86.35**	86.32	86.29	86.24	86.26	86.32	86.24	2
	DINOv2-B	83.05	83.56	83.65	83.60	83.67	**84.08**	83.96	7

F.2 End-to-end allocation curves

In Table 12, the allocation fixed on vision is not the exact perplexity optimum for every language model. For the four models with reservoirs, the retrospective model-level rule misses the measured perplexity optimum in all four; its $k = 5$ choice is the worst of $k \in \{3, 4, 5\}$ for BLOOM-560M and Pythia-1.4B. Nevertheless, all six optima lie at $k = 3$ or $k = 4$, the same narrow region identified in vision. This supports transfer of the useful region and vision-fixed constant, not model-level cross-domain validation of the calibration rule.

F.3 All-Linear W4A4 stress test

The main LLM result (Table 2) deliberately isolates post-GELU activations over the 8-bit base. As separate context, Table 13 quantizes all transformer Linear weights and inputs to four bits for the

Table 11: Calibration-only allocation statistics at LLM post-GELU sites. S_{pos} is the fitted positive clip at the even split. Step 2 rounds the per-model mean prediction; the median columns summarize site-level distributions. The last column substitutes GELU's analytic negative bound.

Model	Sites	Negative mass	Median S_{pos}	Median local k	Local range	Median predicted k	Mean predicted k	Median predicted k (bound-substituted)
BLOOM-560M	24	0.840	1.66	5	3–5	4.76	4.64	4.76
Pythia-1.4B	24	0.870	1.98	4	2–6	4.81	4.61	4.81
Pythia-160M	12	0.799	2.12	4	2–5	4.52	3.91	4.52
Pythia-410M	24	0.834	2.26	4	2–6	4.62	4.21	4.62

Table 12: WikiText-2 perplexity across negative/nonnegative code allocations (lower is better). Compare values only within a row because tokenizers and, for GPT-2, block lengths differ.

Model	$2/14$	$3/13$	$4/12$	$5/11$	Best
Pythia-160M	50.29	**49.08**	50.77	52.56	$3/13$
Pythia-410M	47.98	**45.01**	45.95	50.01	$3/13$
Pythia-1.4B	35.08	**32.68**	33.09	33.35	$3/13$
BLOOM-560M	32.30	28.59	**28.54**	28.71	$4/12$
Phi-2	14.01	12.64	**12.57**	12.74	$4/12$
GPT-2	46.65	**42.02**	42.76	43.82	$3/13$

Table 13: All-Linear W4A4 context on WikiText-2 (perplexity; lower is better). "0.1–99.9%" denotes two-sided percentile clipping on the calibration reservoir; MSE denotes a calibration-reservoir MSE-minimizing static range. SQ is the SmoothQuant-style treatment with $\alpha = 0.5$ and $s_c = \max|X_c|^{0.5} / \max|W_{:,c}|^{0.5}$: it applies X_c/s_c and $W_{:,c}s_c$ at `qkv`, `fc1`, and `fc2`, then recalibrates the smoothed activations; output projections remain plain. These four evaluated rows must not be compared numerically with the site-isolated LLM results, which use the 8-bit base.

Model	FP32	Plain 0.1–99.9%	Plain MSE	SQ 0.1–99.9%	SQ MSE
BLOOM-560M	22.37	100.28	182.47	187.49	149.61
Pythia-1.4B	11.79	2154.87	2187.37	1787.17	1499.32
Pythia-160M	26.65	12551.79	4332.30	386.70	266.45
Pythia-410M	15.95	2174.51	3511.93	678.29	609.64

four evaluated Pythia/BLOOM models, leaving the LM head floating point. This harder operating point remains severely degraded. The SmoothQuant-style rows are an upper bound rather than a zero-overhead deployed baseline: factors before `qkv`/`fc1` can fold into LayerNorm, but a factor before `fc2` cannot be pushed through GELU and would require an explicit per-channel rescale.

F.4 Two-sided-site negative control

In Table 14, applying $4/12$ to the two-sided attention inputs worsens perplexity relative to $8/8$ for all six models. All non-`qkv` components are identical, `qkv` uses four-bit activations, and only the division of the sixteen codes changes. This is cost-matched within the table, but it is not directly comparable to the post-GELU table because it quantizes a different tensor set.

Table 14: WikiText-2 perplexity for the matched two-sided-site control (lower is better). Positive Δ means the unequal allocation is worse. For each model, the test forms paired differences d_i between the two configurations' mean NLLs on the same text blocks, computes $t = \bar{d}/(s_d/\sqrt{n})$, and uses the two-sided normal-approximation tail $p = \operatorname{erfc}(|t|/\sqrt{2})$. Because contiguous blocks need not be independent, this is a descriptive screen rather than a calibrated error rate. Values below 10^{-16} are reported at that uniform threshold.

Model	One range	Even $8/8$	Unequal $4/12$	Δ PPL $(4/12 - 8/8)$	Paired t screen p
Pythia-160M	244.36	110.03	892.63	$+782.60$	$< 10^{-16}$
Pythia-410M	79.40	122.62	3654.98	$+3532.36$	$< 10^{-16}$
Pythia-1.4B	268.94	416.58	965.87	$+549.29$	$< 10^{-16}$
BLOOM-560M	37.20	176.29	567.89	$+391.60$	$< 10^{-16}$
Phi-2	1417.46	19694.92	43572.38	$+23877.46$	$< 10^{-16}$
GPT-2	1104.41	522.80	889.08	$+366.28$	$< 10^{-16}$

Table 15: Per-setting matched-control detail. Each cell reports "treatment minus reference" in top-1 percentage points, followed by the paired exact McNemar p-value; positive Δ favors the treatment. The comparison-specific references are stated in the column headers. Values below 10^{-16} are reported uniformly as $p < 10^{-16}$ rather than as numerical underflow or raw software output. BEiT-B is excluded from the `qkv` control because its functional `qkv` projections remain floating point, as in Table 4.

Dataset	Model	Treatment minus matched reference: Δ top-1 in points (exact McNemar p)		
		$4/12$ at `qkv` vs. one range	Per-site MSE allocation vs. fixed $4/12$	$4/12$ + 3-site A8 vs. fixed $4/12$
ImageNet-1k 224 px	ViT-S	-0.61 ($< 10^{-16}$)	-0.62 (7×10^{-4})	-1.05 (5×10^{-11})
	ViT-B	-0.08 (1×10^{-4})	-0.26 (.063)	-2.97 ($< 10^{-16}$)
	ViT-L	-4.86 ($< 10^{-16}$)	$+0.10$ (.215)	-6.36 ($< 10^{-16}$)
	DeiT-T	-14.33 ($< 10^{-16}$)	$+0.19$ (.273)	$+0.34$ (.007)
	DeiT-S	-3.07 ($< 10^{-16}$)	$+0.71$ (7×10^{-5})	$+0.67$ (8×10^{-7})
	DeiT-B	-0.05 (1×10^{-5})	$+0.13$ (.313)	$+0.21$ (.013)
ImageNet-1k 384 px	ViT-S	-0.17 (5×10^{-8})	-0.71 (3×10^{-5})	-14.28 ($< 10^{-16}$)
	ViT-B	-0.06 (.004)	-0.31 (.016)	-3.05 ($< 10^{-16}$)
	ViT-L	-8.44 ($< 10^{-16}$)	-0.12 (.101)	-6.54 ($< 10^{-16}$)
	DeiT-B	-0.10 (4×10^{-7})	$+0.02$ (.835)	-0.26 (2×10^{-4})
	DeiT3-S	-11.54 ($< 10^{-16}$)	-0.48 (.004)	-0.38 (.022)
	DeiT3-B	-11.59 ($< 10^{-16}$)	-0.12 (.238)	-0.40 (7×10^{-6})
BDD100K 512/518 px	ViT-S	-10.38 ($< 10^{-16}$)	-0.40 (.009)	$+0.03$ (.813)
	ViT-B	-11.96 ($< 10^{-16}$)	-0.11 (.314)	-1.73 (5×10^{-16})
	DeiT-T	-2.06 (1×10^{-14})	-0.21 (.228)	-0.19 (.273)
	DeiT-S	-18.36 ($< 10^{-16}$)	-0.12 (.587)	-0.09 (.557)
	DeiT-B	-9.60 ($< 10^{-16}$)	-0.05 (.738)	-0.06 (.511)
	BEiT-B	—	$+0.14$ (.235)	-0.07 (.572)
	DINOv2-B	$+1.12$ (.047)	$+0.08$ (.551)	$+0.01$ (1.000)

G Analytic Cost and Fake-Quantization Timing

The even split and the $4/12$ allocation both store four bits per activation, use two branch scales, and have identical analytic BitOps and ideal packed size. The timing measurement is intentionally a harness diagnostic. It measures eager FP32 fake-quantization operations around an FP32 model and cannot establish deployed INT4 latency or end-to-end acceleration.

For the ViT-B/384 batch-16 measurement, the median total forward times are 226.91 ms for $8/8$ and 226.52 ms for $4/12$ over 50 repetitions; the 0.17% difference is smaller than the scope of a deployment claim. CUDA-event timing used an RTX A5000, PyTorch 2.7.1 with CUDA 12.6, 10 model warmups, and a full device synchronization per repetition.

H Matched Controls

For the three-site A8 control, sites were selected independently for each model–resolution setting rather than reused at common relative depths. Using only the unlabeled calibration reservoir, we ranked post-GELU inputs by descending $P_{99.9}(|a|)/P_{90}(|a|)$ and promoted the top three `mlp.fc2` inputs; no sensitivity or validation accuracy entered the selection. The zero-based block indices were: ImageNet-1k/224, ViT-S/B $[8, 9, 10]$, ViT-L $[19, 20, 21]$, DeiT-T/S $[7, 8, 9]$, and DeiT-B $[8, 9, 10]$; ImageNet-1k/384, ViT-S/B, DeiT-B, and DeiT3-B $[8, 9, 10]$, ViT-L $[19, 20, 21]$, and DeiT3-S $[0, 9, 10]$; BDD100K, ViT-S/B and DeiT-B $[8, 9, 10]$, DeiT-T $[0, 8, 9]$, DeiT-S $[7, 8, 9]$, BEiT-B $[0, 10, 11]$, and DINOv2-B $[6, 8, 10]$. Promoted sites used 99.99th-percentile A8 ranges.

Each control is evaluated at the deployed $4/12$ allocation. Treatment and reference come from the same job and therefore share the model, calibration draw, image order, and evaluation code. The `qkv`

control asks whether adding $4/12$ at a two-sided site helps relative to one range. The per-site selector chooses each post-GELU allocation by calibration reconstruction MSE. The last control adds 8-bit activations at three selected post-GELU sites on top of the fixed $4/12$ allocation. Table 4 gives the aggregate results; Table 15 gives the per-setting paired differences underlying those aggregates.