PATHBALANCE: DIAGNOSING PATH-LOAD IMBALANCE IN DENSE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Dense language models expose no explicit routing distribution, yet fine-tuning can concentrate their internal computation in a small subset of attention heads, MLP neurons, and residual-stream dimensions. We study this phenomenon as *path-load imbalance*, measuring activation strength (A), gradient pressure (G), and activation-gradient attribution (C). On Intermediate Algebra, after the best checkpoint, the top 2% of MLP neurons increase their share of attribution load from 15% to 72%, and the top 2% of residual dimensions from 15% to 36%. Higher residual-C exposure predicts poorer retention of previously correct answers, and suppressing train-hot MLP neurons reduces accuracy from 32.7% to 1.5%. Building on these findings, we introduce **PATHBALANCE**, a load-aware dropout method for full fine-tuning that measures the current load before every update and masks the top-2% hot units while the remaining network is trained with cross-entropy. Across three math benchmarks and five `Qwen3`/`Qwen3.5` models, PATHBALANCE improves over SFT by up to 16 percentage points on GSM8K and 50 on MATH, without changing the inference architecture. Masking the same number of randomly chosen neurons yields no gain, and a matched-epoch analysis shows that PATHBALANCE shifts attribution load from overloaded neurons to previously cold ones. The code is available here.

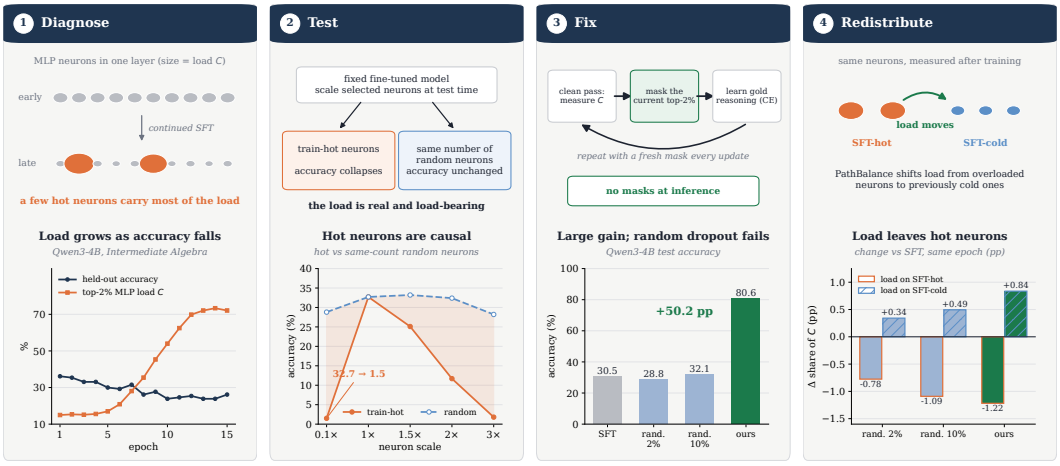


Figure 1: PATHBALANCE at a glance (`Qwen3-4B`). (1) Load concentrates on a few MLP neurons as accuracy falls (Intermediate Algebra). (2) Scaling these neurons collapses accuracy; random neurons do not. (3) Masking hot units during training gains 50 points; random dropout does not. (4) Load moves from SFT-hot to SFT-cold neurons. Targeting the few units that carry the load fixes the problem, and the model learns to use previously cold units instead.

1 INTRODUCTION

Supervised fine-tuning (SFT) is the standard way to adapt a pretrained language model to downstream tasks (Wei et al., 2022; Ouyang et al., 2022), including mathematical reasoning via worked

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.

solutions (Yu et al., 2024). On small datasets, however, SFT is brittle: fine-tuning overwrites earlier abilities (Kirkpatrick et al., 2017; Luo et al., 2025), and in our runs held-out accuracy peaks early, then declines as the model loses answers it once got right (Section 3.5). Remedies such as early stopping, weight decay, or dropout (Srivastava et al., 2014) act on the model as a whole; none asks *where* in the network the fine-tuning signal goes.

Sparse mixture-of-experts models make this question explicit: a router with an auxiliary balancing loss keeps the computation from collapsing onto a few overloaded experts (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022). Dense Transformers have no router, yet a few heads, feed-forward neurons, and residual dimensions carry a disproportionate share of their function or magnitude (Michel et al., 2019; Dai et al., 2022; Dettmers et al., 2022; Sun et al., 2024). We therefore ask: *does SFT concentrate the loss-relevant computation of a dense model onto a few units, and does this concentration explain the loss of held-out accuracy?*

We call this phenomenon *path-load imbalance* and measure MLP neurons, residual dimensions, and attention heads with A, G, and C, taking the first-order attribution C (Nanda, 2023; Kramár et al., 2024) as a unit's load. Fine-tuning `Qwen3-4B` on two MATH subjects reveals four regularities (Section 3): (i) on Intermediate Algebra, after the best checkpoint, the top 2% of MLP neurons grow from 15% to 72% of the load and the top 2% of residual dimensions from 15% to 36%, while activations stay flat; (ii) these hot units are causally load-bearing, as scaling them collapses accuracy (MLP from 32.7% to 1.5%, residual to 0%) far more than scaling random units; (iii) high residual load predicts forgetting (retention 30.4% vs. 66.7%); and (iv) the harm comes from *newly* hot neurons, while neurons hot from the start are useful.

Hot units should therefore not be suppressed permanently; instead, the network should not rely on whichever units are overloaded *now*. PATHBALANCE is a load-aware dropout for full fine-tuning: before every update, an unmasked pass measures the current C of each unit, the top 2% (12.5% of heads) are masked, and the remaining network is trained on the same data with ordinary cross-entropy (CE). It adds no auxiliary loss or teacher and leaves inference unchanged. Across GSM8K and two MATH subjects (Cobbe et al., 2021; Hendrycks et al., 2021) and five `Qwen3`/`Qwen3.5` models (Qwen Team, 2025; 2026), it improves over matched SFT in almost every setting, by up to 16 percentage points on GSM8K and 50 on MATH, whereas masking the same number of random neurons, or five times as many, gives no comparable gain. Our contributions are:

- **Diagnosis:** a component-resolved account of path-load imbalance, showing that load concentrates, hot units are causally load-bearing, and late concentration predicts forgetting (Section 3).
- **Method:** PATHBALANCE, which re-measures load before every update and masks the currently hot units, trained only with masked gold CE (Section 4).
- **Evidence:** gains across five models and three datasets, with controls and an ablation showing that load-based selection drives the gains and moves load onto cold neurons (Sections 4.4–4.6).

2 RELATED WORK

Interpretability research shows that Transformer computation is localized. Individual attention heads specialize (Clark et al., 2019; Voita et al., 2019) and many can be pruned (Voita et al., 2019; Michel et al., 2019), feed-forward layers act as key-value memories whose neurons store specific knowledge (Geva et al., 2021; Dai et al., 2022), small circuits implement specific behaviors (Elhage et al., 2021; Wang et al., 2023), and dictionary learning recovers interpretable features (Bricken et al., 2023). Causal tracing, activation patching, and automated circuit discovery test which components matter (Meng et al., 2022; Zhang & Nanda, 2024; Conmy et al., 2023), while gradient-based attribution (Sundararajan et al., 2017) and first-order attribution patching (Nanda, 2023; Kramár et al., 2024) give inexpensive estimates. Large language models also develop a few outlier or massive activation dimensions that dominate the residual stream (Dettmers et al., 2022; Sun et al., 2024). Our C statistic belongs to this attribution family and is not an exact causal decomposition.

Fine-tuning on a narrow task can overwrite previously acquired abilities, a form of catastrophic forgetting (Kirkpatrick et al., 2017; Luo et al., 2025), and neuron-level SFT updates only a selected subset of neurons (Xu et al., 2025). For mathematical reasoning, models are fine-tuned on supplied or bootstrapped solutions to GSM8K and MATH problems (Cobbe et al., 2021; Hendrycks et al.,

2021; Yu et al., 2024), or distill rationales from a teacher model (Hsieh et al., 2023; Shridhar et al., 2023).

Dropout discourages co-adaptation by removing random units during training (Srivastava et al., 2014). Adaptive dropout learns input-dependent keep probabilities (Ba & Frey, 2013), and targeted dropout stochastically drops low-magnitude weights or units so that the network tolerates later pruning (Gomez et al., 2019). Closer to our work, guided dropout, excitation dropout, and RSC drop highly important units or features of vision classifiers by learned strength, saliency, or gradient (Keshari et al., 2019; Zunino et al., 2021; Huang et al., 2020). Other methods reallocate underused capacity (Qiao et al., 2019) or counter competition among predictive features (Pezeshki et al., 2021).

Our work differs from these lines in three ways. First, instead of analyzing a fixed trained model, we measure how fine-tuning redistributes loss-relevant load over the units of a dense model, and show that the resulting hot units are causally load-bearing and that their late concentration predicts held-out forgetting. Second, unlike random, adaptive, or magnitude-based dropout, which remove random or unimportant units, and like guided dropout, excitation dropout, and RSC, PATHBALANCE removes important units, but it ranks them by gold-loss attribution C over the whole update window, re-measured before every step, rather than by a learned strength or a per-example saliency or gradient, and masks MLP neurons, residual dimensions, and heads of pretrained language models during full fine-tuning. A same-size random control and the redistribution analysis show that this selection rule, not dropout itself, drives the gain and moves load onto previously cold units. Third, unlike mixture-of-experts balancing, it requires no router, experts, or auxiliary loss, and unlike neuron-level fine-tuning, it does not restrict which parameters are updated: all parameters remain trainable, the mask only constrains which units may carry the loss at a given step, and inference uses the unmodified model.

3 PATH-LOAD IMBALANCE

We define load and show that it concentrates, is causally load-bearing, and predicts forgetting.

3.1 MEASURING LOAD: ACTIVATION, GRADIENT, AND ATTRIBUTION

Let f_θ be a dense decoder-only model and (x, y) a prompt with its gold solution. A unit u in layer ℓ is an MLP channel (the input of the down projection), a residual-stream coordinate, or a complete attention output-head vector. For a scalar activation $a_{\ell u,t}$ at position t and its gold-CE gradient $g_{\ell u,t} = \partial\mathcal{L}/\partial a_{\ell u,t}$, we define

$$A_{\ell u} = \mathbb{E}_t|a_{\ell u,t}|, \qquad G_{\ell u} = \mathbb{E}_t|g_{\ell u,t}|, \qquad C_{\ell u} = \mathbb{E}_t|a_{\ell u,t}\, g_{\ell u,t}|. \tag{1}$$

For heads, A and G use ℓ_2 norms and $C_{\ell u} = \mathbb{E}_t|\langle a_{\ell u,t}, g_{\ell u,t}\rangle|$, with the absolute value taken after summing over the head's coordinates. The expectation covers every gold-prediction position of the supplied reasoning and final answer; prompt and padding positions are excluded.

The three channels answer different questions. *Activation* A asks which units fire; it needs only a forward pass, but a strongly active unit may be irrelevant to the loss. *Gradient* G asks where the loss applies learning pressure; a unit can receive a large gradient while being nearly inactive. *Attribution* C asks which units are both active and loss-sensitive: $|a\, g|$ is the first-order estimate of how much the loss would change if the unit's contribution were removed. C is therefore our definition of *load*, while A and G tell us whether a change in load comes from stronger firing or from stronger learning pressure. Appendix Table 5 summarizes their roles, and Section 3.3 shows empirically that C carries the clearest signal.

To quantify concentration, for a channel $X \in \{A, G, C\}$ we normalize within each layer and component,

$$p^X_{\ell u} = \frac{X_{\ell u}}{\sum_v X_{\ell v} + \epsilon}, \qquad \mathcal{S}^X_\ell = \mathrm{TopK}_u\, X_{\ell u}, \qquad \mu^X_\ell = \sum_{u \in \mathcal{S}^X_\ell} p^X_{\ell u}. \tag{2}$$

The top-set mass μ is the share of a layer's load carried by its hot set $\mathcal{S}$; under a uniform distribution the top 2% of units would carry exactly 2%. We average μ with equal weight over layers so that large-magnitude layers do not dominate. *Training concentration* is the mass of the

Table 1: Diagnostic evidence (`Qwen3-4B`); (a), (c): Intermediate Algebra / Precalculus. (a) Top-k load mass (%), best checkpoint → after full training (heads: top 3 of 32; MLP, residual: top 2%). (b) Generation-time scaling of train-hot vs. same-count random units, pooled over both subjects (heads: Int. Alg.). (c) Retention (%) of samples correct at epoch 1, by exposure (colored: $p < 0.05$ on Int. Alg.). Red: harmful or growing load; green: unchanged or beneficial.

(a) Load concentrates after the best checkpoint

Channel	Heads	MLP	Residual
C	23 → **26** / 23 → **29**	15 → **72** / 20 → **68**	15 → **36** / 15 → **55**
A	19 → **18** / 18 → **18**	11 → **11** / 11 → **11**	12 → **10** / 11 → **10**
G	12 → **12** / 12 → **12**	3 → **12** / 5 → **13**	4 → **18** / 5 → **20**

(b) Hot units are causally load-bearing

Component	Intervention	Baseline	Hot units	Random units
MLP	Suppress 0.1×	134/410 = 32.7%	**6/410 = 1.5%**	**118/410 = 28.8%**
MLP	Amplify 2.0×	134/410 = 32.7%	**48/410 = 11.7%**	**133/410 = 32.4%**
Residual	Suppress 0.1×	134/410 = 32.7%	**0/410 = 0.0%**	**95/410 = 23.2%**
Residual	Amplify 2.0×	134/410 = 32.7%	**0/410 = 0.0%**	**94/410 = 22.9%**
Heads	Suppress 0.1×	39/84 = 46.4%	**7/84 = 8.3%**	**27/84 = 32.1%**
Heads	Amplify 2.0×	39/84 = 46.4%	**11/84 = 13.1%**	**34/84 = 40.5%**

(c) Residual load and head gradient predict forgetting

Channel	Component	High exposure	Low exposure	p
C	Heads	37.7 / 68.9	59.7 / 80.0	0.118 / 0.514
C	MLP	50.7 / 75.6	47.2 / 73.3	0.836 / 0.999
C	Residual	**30.4 / 55.6**	**66.7 / 93.3**	0.009 / 0.009
A	Heads	50.7 / 66.7	47.2 / 82.2	0.833 / 0.353
A	MLP	46.4 / 66.7	51.4 / 82.2	0.752 / 0.364
A	Residual	**31.9 / 60.0**	**65.3 / 88.9**	0.017 / 0.059
G	Heads	**26.1 / 62.2**	**70.8 / 86.7**	0.001 / 0.115
G	MLP	39.1 / 62.2	58.3 / 86.7	0.174 / 0.114
G	Residual	**34.8 / 62.2**	**62.5 / 86.7**	0.047 / 0.115

training distribution on its own hot set; *held-out exposure* is how much of a held-out example's load falls on the training-defined hot set; and *top-set overlap* $|\mathcal{S}_{\text{train}} \cap \mathcal{S}_x|/|\mathcal{S}_{\text{train}}|$ compares the membership of two independently ranked hot sets. We also report the effective number of units, $N_{\text{eff}} = \exp(-\sum_u p_u \log p_u)$.

3.2 Diagnostic Setup

We fine-tune `Qwen3-4B` for 15 epochs on two MATH subjects (Hendrycks et al., 2021), Precalculus and Intermediate Algebra, with full fine-tuning, per-sample load logging, and a checkpoint after every epoch. For attention we track the top 3 of 32 heads per layer; for MLP and residual units, the top 2% per layer (195 of 9728 MLP neurons). Unless stated otherwise, "hot" means the top units by training-set C within a layer and checkpoint. The held-out cohorts are validation-sized (Appendix Table 6). Causal probes intervene only during generation and never modify the checkpoint weights; each is compared with a same-count random control (full protocol in Appendix G).

3.3 Load Concentrates on a Few Units

Attribution is the most informative channel: across components, C exposes the structure that A alone misses (Appendix Table 5). Activation mass stays essentially flat after the best checkpoint, whereas attribution and gradient mass grow, most strongly in MLP and residual units. The remaining analyses therefore rank units by C.

Load is already uneven at the best Intermediate Algebra checkpoint: the top 2% of MLP neurons carry 15% of MLP C, over seven times their uniform share, and the top 2% of residual dimensions carry 15%. With continued training this grows to 72% for MLP and 36% for residual units, while activation mass is unchanged; Precalculus shows the same growth (Table 1a). Fine-tuning past the best checkpoint thus funnels the loss-relevant computation through an increasingly small set of units.

3.4 HOT UNITS ARE CAUSALLY LOAD-BEARING

To test whether the hot MLP neurons actually carry the computation, we scale the train-hot MLP-C neurons at the input of the down projection during generation and compare with the same number of random non-hot neurons (Table 1b). Suppressing the hot neurons collapses accuracy from 32.7% to 1.5%: 131 baseline-correct outputs become wrong and only 3 become correct, whereas the random control stays at 28.8%. Amplifying them also hurts (11.7%), and the harm grows smoothly with the amplification scale while the random control stays flat (Figure 1, panel 2; the 3.0$\times$ point uses 280 of the 410 pairs). The hot set is both necessary and fragile: removing it breaks the model, and pushing more signal through it breaks it as well.

The same test on the residual stream scales the train-hot residual-C dimensions (51 of 2560 per layer) at the output of every decoder layer, on the same 410 sample/checkpoint units (Table 1b). Suppressing or amplifying them drives accuracy from 32.7% to 0% in both directions: every baseline-correct output becomes wrong. The same number of random dimensions leaves 23.2% and 22.9%. Random residual perturbations also cost about ten points, so the residual stream is sensitive overall, yet hot dimensions matter far more on every subject and checkpoint (Appendix Table 4).

Attention load is also uneven: the top 3 heads per layer (9.4% of heads) carry 23% of train C at the best checkpoint and 26% after full training on Intermediate Algebra (23% to 29% on Precalculus; Table 1a). Perturbing these heads is far more damaging than perturbing random heads (Table 1b; pooled hot-versus-random $p = 0.0001$ and $p = 0.0002$ over 84 sample/checkpoint units), but the mechanism differs from MLP overload: hot-head perturbations mostly produce unterminated, repetitive outputs (Appendix A). Hot heads carry the load needed for coherent generation.

3.5 LOAD GROWTH PREDICTS FORGETTING

Over training, held-out accuracy falls while load concentration rises. On Intermediate Algebra the best checkpoint is epoch 1; from epoch 1 to 15, held-out accuracy falls from 47/130 to 34/130 while MLP-C concentration rises from 15% to 72% (Figure 1, panel 1; Table 1a); the two are strongly anti-correlated across epochs ($r = -0.879$, permutation $p = 0.0001$). Table 1a shows that this growth appears in C and G but not in A, and high MLP-C exposure also predicts more next-epoch flips from correct to wrong (Appendix A).

Residual load also predicts forgetting, and so does head gradient exposure. Among the 47 Intermediate Algebra samples that are correct at epoch 1, those with high residual-C exposure retain their correct answer far less often than those with low exposure (30.4% vs 66.7%, $p = 0.0092$; Table 1c), and the effect holds for all three residual channels. Head G exposure is an even stronger predictor (26.1% vs 70.8%, $p = 0.0011$), whereas head C and MLP exposure do not predict retention at the end of training. Precalculus replicates the residual-C effect (55.6% vs 93.3%, $p = 0.009$; Table 1c).

3.6 NEWLY HOT NEURONS ARE THE HARMFUL PART

Not every hot neuron is harmful. At epoch 15, scaling the current top-2% MLP-C neurons down by 0.8 worsens teacher-forced loss by $+0.0121$, but splitting this set by history resolves the effect: neurons that were already hot at epoch 1 (stable-hot) are useful, since scaling them down raises the loss by $+0.0567$, whereas neurons that became hot later (new-hot) are harmful, since scaling them down lowers the loss by 0.0505. The same holds in generation on 90 high-risk regressed sample/checkpoint pairs: scaling only the new-hot neurons by 0.8 raises accuracy from 13/90 to 24/90, with 16 rescues and 5 harms, while a same-count random control reaches only 15/90.

Because useful stable-hot neurons coexist with harmful late-emerging ones, permanently suppressing hot units is not the answer; instead, the remaining network should learn to carry the computation without them, which motivates PATHBALANCE.

4 PATHBALANCE: LOAD-AWARE TRAINING

PATHBALANCE removes the currently hot units at every update, so the rest of the network must learn the correct solution without them (Figure 2).

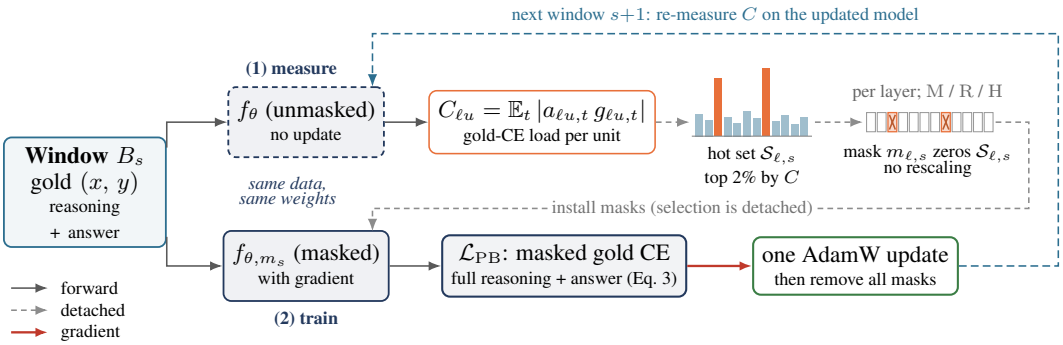


Figure 2: One PATHBALANCE training update. (1) An unmasked pass measures the load C of every unit, and the top 2% per layer and site (12.5% of heads) are masked without rescaling. (2) The same window is trained with masked CE; after one AdamW update the masks are removed and C is measured again. Inference uses the unmasked model.

4.1 METHOD

Each update begins with a fresh measurement of load. Let the window B_s contain all microbatches of optimizer update s, with T_s supervised target tokens, where $\mathcal{T}(y)$ denotes the supervised target positions of y. We first run the *unmasked* model on B_s, backpropagate the gold CE, and accumulate C (Eq. 1); no optimizer step is taken and the resulting parameter gradients are discarded. PATHBALANCE can mask hot units in three model parts, and we name each masking setting by the part it masks: **M** masks MLP neurons, **R** residual dimensions, and **H** attention heads; combinations such as M+R mask several parts at once. Throughout, "MLP", "residual", and "heads" refer to the model parts themselves, and M, R, and H only to masking settings. For each targeted part and layer, the hot set $\mathcal{S}_{\ell,s}$ consists of the top $k = \max(1, \lfloor 0.02d \rfloor)$ units of width d by current C, or the top 12.5% of output heads. The hot set is always computed from the current model and batch, with no moving average or fixed SFT hot set.

Training then uses masked gold supervision: a binary mask $m_{\ell,s}$ zeros the entire hot set ($m_s = \{m_{\ell,s}\}$ collects the masks of all layers and targeted parts), shared across the examples and positions of the window and without inverted-dropout rescaling. On the *same data* we then optimize only

$$\mathcal{L}_{\mathrm{PB}}(\theta; m_s) = -\frac{1}{T_s} \sum_{(x,y)\in B_s} \sum_{t\in\mathcal{T}(y)} \log p_{\theta,m_s}(y_t \mid x, y_{<t}), \tag{3}$$

the teacher-forced CE over the whole supplied reasoning and answer. Mask selection is detached, there is one AdamW update per window, and the masks are recomputed for the next window (pseudocode in Appendix Algorithm 1). There is no auxiliary loss, no teacher imitation, and no change at inference: validation and test use the unmasked model. The extra measurement pass increases the cost of a training step by about 2.0–2.3× (Appendix D.3).

This gives other units a learning signal they lack under SFT, and because the mask follows the current load, newly hot units (Section 3.6) are removed as soon as they emerge.

4.2 INTERVENTION SITES

Each decoder layer ℓ updates the residual stream h_ℓ in two steps (normalization omitted), $u_\ell = h_\ell + W_O^\ell[o_{\ell,1}; \ldots; o_{\ell,n_h}]$ and $h_{\ell+1} = u_\ell + W_{\mathrm{down}}^\ell z_\ell$, where $o_{\ell,j}$ is the output vector of attention head j and $z_\ell = \phi(W_{\mathrm{gate}}^\ell u_\ell) \odot W_{\mathrm{up}}^\ell u_\ell$ is the gated MLP activation. Each mask zeros the hot entries of one of these quantities (illustrated in Appendix Figure 4):

- **M:** hot entries of z_ℓ, which removes those neurons' columns of W_{down} from this layer's MLP update and nothing else.
- **H:** hot head outputs $o_{\ell,j}$ before W_O, which removes those heads' entire contribution to the layer (for the hybrid `Qwen3.5`, see Appendix B.2).

- **R:** hot coordinates of $h_{\ell+1}$ after both additions. Because $h_{\ell+1}$ also carries what earlier layers wrote, **R** removes that information too and is broader than a same-size **M** or **H**.

Each setting masks its hot units in every layer at once, which matches (within one point) or beats random-layer masking (Appendix B.1).

4.3 Experimental Setup

We fine-tune `Qwen3-0.6B`, `Qwen3-1.7B`, `Qwen3-4B`, `Qwen3.5-0.8B`, and `Qwen3.5-2B` (Qwen Team, 2025; 2026) from their released checkpoints (Appendix Table 8; checkpoint links in Appendix F), with thinking disabled. We use full fine-tuning with all parameters trainable. The benchmarks are GSM8K (Cobbe et al., 2021) and the Intermediate Algebra and Precalculus subjects of MATH (Hendrycks et al., 2021). A fixed 10% of each official training set is held out for validation, and the official test sets are used only for the final evaluation.

SFT and PathBalance share every optimization setting within a comparison (AdamW, learning rate 2×10^{-5}; Appendix Table 9). Validation accuracy is computed every 60 steps, training stops after ten checks without improvement, and the test set is evaluated once, at the best checkpoint.

Evaluation uses greedy vLLM generation (Kwon et al., 2023) without masks; GSM8K answers are matched numerically, MATH answers with `math-verify` (Kydlíček, 2025) (Appendix F).

4.4 Main Results

Table 2 reports test accuracy for SFT and every component setting on all five models and three datasets. Paired uncertainty is in Appendix Table 10, and further results in Appendix D.

Table 2: Test accuracy (%) across the three datasets. All rows use full fine-tuning. Parentheses show signed percentage-point differences from the matched SFT baseline within each dataset and model. Each setting masks the full hot pool of the listed components (top-2% MLP neurons / residual dimensions; 12.5% of heads) in all layers.

Setting	Q3-0.6B	Q3-1.7B	Q3.5-0.8B	Q3.5-2B	Q3-4B
GSM8K					
SFT	47.61	79.38	46.25	63.08	92.95
M	63.00 (+15.39)	81.12 (+1.74)	55.88 (+9.63)	77.33 (+14.25)	93.63 (+0.68)
R	63.61 (+16.00)	80.89 (+1.52)	55.04 (+8.79)	77.26 (+14.18)	92.87 (-0.08)
H	61.79 (+14.18)	80.74 (+1.36)	54.89 (+8.64)	75.66 (+12.59)	93.18 (+0.23)
M+R	63.00 (+15.39)	80.89 (+1.52)	56.56 (+10.31)	77.18 (+14.10)	92.72 (-0.23)
M+H	62.09 (+14.48)	80.74 (+1.36)	55.72 (+9.48)	77.56 (+14.48)	93.78 (+0.83)
M+R+H	62.62 (+15.01)	80.89 (+1.52)	56.10 (+9.86)	77.33 (+14.25)	93.18 (+0.23)
Intermediate Algebra					
SFT	9.97	15.95	9.52	16.83	30.45
M	31.67 (+21.71)	55.04 (+39.09)	13.62 (+4.10)	32.12 (+15.28)	80.62 (+50.17)
R	34.22 (+24.25)	57.59 (+41.64)	19.05 (+9.52)	50.06 (+33.22)	79.62 (+49.17)
H	32.12 (+22.15)	56.59 (+40.64)	17.83 (+8.31)	40.86 (+24.03)	79.96 (+49.50)
M+R	34.77 (+24.81)	56.59 (+40.64)	19.27 (+9.75)	47.62 (+30.79)	80.29 (+49.83)
M+H	32.23 (+22.26)	57.48 (+41.53)	21.48 (+11.96)	50.28 (+33.44)	79.96 (+49.50)
M+R+H	32.78 (+22.81)	57.03 (+41.09)	21.48 (+11.96)	46.62 (+29.79)	80.07 (+49.61)
Precalculus					
SFT	9.71	18.13	9.71	15.20	34.62
M	28.94 (+19.23)	51.83 (+33.70)	12.45 (+2.75)	16.48 (+1.28)	83.88 (+49.27)
R	33.70 (+23.99)	59.52 (+41.39)	23.44 (+13.74)	50.18 (+34.98)	81.32 (+46.70)
H	28.02 (+18.32)	53.85 (+35.71)	17.03 (+7.33)	37.00 (+21.79)	78.21 (+43.59)
M+R	34.80 (+25.09)	60.26 (+42.12)	24.91 (+15.20)	58.42 (+43.22)	82.05 (+47.44)
M+H	37.00 (+27.29)	55.49 (+37.36)	27.66 (+17.95)	52.56 (+37.36)	80.59 (+45.97)
M+R+H	37.55 (+27.84)	57.51 (+39.38)	31.32 (+21.61)	58.24 (+43.04)	81.68 (+47.07)

PathBalance improves over SFT almost everywhere. On MATH, every model improves under every component setting. The largest gain on each subject is 50.17 points on `Qwen3-4B` Intermediate Algebra (30.45% to 80.62% with M) and 49.27 points on `Qwen3-4B` Precalculus (34.62% to 83.88% with M), and `Qwen3-1.7B` gains more than 33 points in every MATH cell. On GSM8K, gains reach 16.00 points for `Qwen3-0.6B` (R) and 14.48 points for `Qwen3.5-2B` (M+H); they

are smaller where SFT is already strong (`Qwen3-1.7B`, $+1.4$ to $+1.7$) and within one point for `Qwen3-4B`, whose SFT baseline is already 92.95%.

The best setting depends on the model: on `Qwen3.5-2B` Precalculus, M gains 1.28 points, whereas R gains 34.98 and M+R 43.22. R and the combined settings are consistently strong on the hybrid `Qwen3.5` models, while `Qwen3-4B` benefits similarly from every component. Masking the whole decoder, not only MLP neurons, therefore matters.

4.5 Ablation: Where Does the Load Go?

To test whether selecting units by load is what matters, and where the load goes, we retrain `Qwen3-4B` on Intermediate Algebra with the main 25-epoch learning-rate schedule and validation-based checkpoint selection: SFT, random dropout of 2% or 10% of MLP neurons, hot 2% dropout, and three variants that add the removed activation energy back to the remaining or least-loaded neurons. For each run we report test accuracy and measure clean MLP C at the same epoch on the same 32 held-out questions, comparing against SFT's own hot (top-2%) and cold (bottom-10%) neurons in each layer (Table 3).

Table 3: Ablation and load redistribution (`Qwen3-4B`, Intermediate Algebra). Left: test accuracy. Right: clean MLP load C at the same epoch for every run, on 32 held-out questions. Top-2%: the model's own top-2% share of C; overlap: fraction of its hot neurons that are SFT-hot; Δ hot/cold: change from same-epoch SFT in the C share of SFT's top-2% / bottom-10% neurons; dist.: total-variation distance from SFT's C distribution.

	Test accuracy		Load at the same epoch				
Run	Test (%)	Δ SFT (pp)	Top-2% C (%)	Overlap (%)	Δ hot (pp)	Δ cold (pp)	Dist. (%)
SFT	30.45	–	12.21	100.0	0	0	0
Controls							
Random 2%	28.79	-1.66	12.18	75.6	-0.78	+0.34	8.49
Random 10%	32.12	+1.66	12.36	68.3	-1.09	+0.49	10.89
Hot 2% (ours)	80.62	+50.17	13.10	59.8	-1.22	+0.84	13.62
Ablation: explicit amplification							
+ amplify remaining	80.62	+50.17	13.50	55.6	-1.35	+1.05	15.62
+ amplify bottom 5%	80.40	+49.94	14.16	57.2	-0.73	+0.60	15.69
+ amplify bottom 20%	82.06	+51.61	14.12	56.4	-0.85	+0.59	16.23

Selecting units by load is what matters. Hot 2% dropout gains 50.17 points over SFT, whereas random dropout of the same number of neurons, or of five times as many, stays within two points of SFT (Table 3, left). The gain comes from *which* neurons are masked, not from dropout itself.

Hot dropout also moves load from SFT's hot neurons to its previously cold ones: at the same epoch, SFT's hot neurons carry 1.22 points less of the load and its cold neurons 0.84 points more (Table 3, right), and random dropout moves much less load onto SFT's cold neurons, even at 10%. Hot dropout also changes more neuron identities than either random control (lower overlap).

Amplifying the remaining neurons after masking keeps the accuracy of hot dropout and gives the largest shift from hot to cold neurons of all runs. All hot-based variants reach 80.4–82.1%, so the benefit is robust to how the removed energy is handled, and we keep the simpler method. The model's own top-2% share stays close to SFT's, so PathBalance moves load across neurons rather than flattening it.

4.6 Why Does SFT Fall Behind?

The ablation runs also explain the size of the gap (Figure 3). The four runs share the data, schedule, and seed; they differ only in the training mask.

The gap opens immediately. At the first check (step 60), hot dropout already reaches 88.5%, whereas SFT and random dropout reach 28.5–33.8%; the ordering then holds throughout training (Figure 3a). Hot dropout thus follows a different trajectory from the start, beyond avoiding late overfitting.

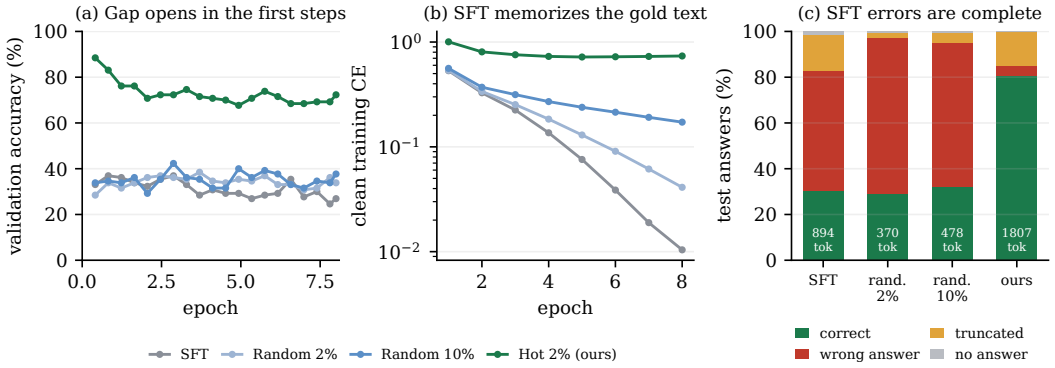


Figure 3: Why SFT falls behind (`Qwen3-4B`, Intermediate Algebra, four of the runs of Table 3). (a) Validation accuracy every 60 steps. (b) Clean (unmasked) training CE on the gold solutions, averaged per epoch. (c) Outcome of every test answer at the selected checkpoint, with mean length (exclusive categories, unlike Table 11).

With continued training, SFT memorizes the gold text: measured without masks, its training CE on the supplied solutions falls from 0.53 to 0.010, and random dropout also falls far below hot dropout, which stays near 0.73 (Figure 3b). Removing the units that carry the fit at every update prevents this memorization, which is why SFT never recovers; the gap itself is already present at the first validation checks, where both models reach their best checkpoints.

Finally, SFT fails by reasoning incorrectly, not by format: most of its failures are complete answers with a wrong result (474 of 903 test questions, against 38 for PATHBALANCE), while truncation and missing answers are secondary (Figure 3c; all models in Appendix I). SFT and random dropout also write much shorter answers than PATHBALANCE, closer to the short supplied solutions, whereas PATHBALANCE keeps writing full derivations. Imitating the concise training solutions thus shortens the model's reasoning, and masking the hot units prevents it.

5 LIMITATIONS

Our study is limited to full fine-tuning of `Qwen3` and `Qwen3.5` models on mathematical reasoning, with one seed and one shared recipe per comparison; results on other domains, model families, seeds, and models beyond 4B parameters remain to be established. We use one masking fraction (2% of units, 12.5% of heads) for all models without per-model tuning, so the reported gains may not be the best achievable. The measurement pass increases the cost of a training step by about 2.0–2.3×; refreshing masks every few updates may retain most of the benefit at lower cost. Our redistribution analysis shows that load moves from SFT's hot neurons to previously cold ones; which functions these neurons take on is left for mechanistic study. We evaluate with greedy, non-thinking generation under fixed token budgets; sampling or thinking-mode decoding may differ (Appendix E).

6 CONCLUSION

We presented PATHBALANCE as both a diagnostic of internal dependence and a training intervention. Activation-gradient measurements reveal concentrated, component-specific load that activation magnitude alone misses; controlled probes show that hot MLP neurons, residual dimensions, and heads are disproportionately load-bearing, and that newly hot MLP units are the harmful part. Masking the currently hot units during fine-tuning, and learning from correct reasoning and answers without them, improves over matched SFT almost everywhere across five models and three datasets, with no change to the inference architecture. Compared with random dropout of the same size, it moves more load onto previously cold neurons, redistributing load rather than flattening it.

Future work includes larger models, other domains and model families, and reinforcement-learning fine-tuning; cheaper mask refreshing every few updates; per-model and per-part masking rates; and a mechanistic study of which functions the newly loaded units take on.

AI USE STATEMENT

Generative AI tools were used to assist with drafting and editing the manuscript, format LaTeX content, and support the design of scientific figures. All AI-assisted suggestions and outputs were reviewed by the authors for correctness, relevance, and originality. The authors take full responsibility for the final methodology, experiments, claims, and presentation.

REPRODUCIBILITY STATEMENT

The load measurement and diagnostic protocol are specified in Section 3 and Appendices A and G. The PATHBALANCE objective, hot-set selection, and masking sites are given in Section 4, with pseudocode in Appendix Algorithm 1. Section 4.3 and Appendix C give models, data splits, optimization, and checkpoint selection, and Appendix F documents checkpoint identities, answer extraction, and scoring. Code and configurations for all reported runs are available via the link in the abstract.

ETHICS STATEMENT

This study fine-tunes publicly released models on public mathematical reasoning benchmarks and involves no human subjects or personal data. Accuracy on these benchmarks does not establish reliability for educational or other consequential deployment. We report generation failures such as truncation and missing answers (Appendix I) to avoid overstating reliability, and the supplied reference solutions may themselves contain errors.

REFERENCES

Jimmy Ba and Brendan Frey. Adaptive dropout for training deep neural networks. In *Advances in Neural Information Processing Systems*, volume 26, 2013. URL https://papers.neurips.cc/paper/5032-adaptive-dropout-for-training-deep-neural-networks.pdf.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Chris Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? An analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 276–286, 2019.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, 2023.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, 2022.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

Aidan N. Gomez, Ivan Zhang, Siddhartha Rao Kamalakara, Divyam Madaan, Kevin Swersky, Yarin Gal, and Geoffrey E. Hinton. Learning sparse networks using targeted dropout. *arXiv preprint arXiv:1905.13678*, 2019. URL https://arxiv.org/abs/1905.13678.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2021.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! Outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL*, 2023.

Zeyi Huang, Haohan Wang, Eric P. Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *European Conference on Computer Vision*, 2020.

Rohit Keshari, Richa Singh, and Mayank Vatsa. Guided dropout. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4065–4072, 2019. doi: 10.1609/aaai.v33i01.33014065.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. AtP*: An efficient and scalable method for localizing LLM behaviour to components. *arXiv preprint arXiv:2403.00745*, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165.

Hynek Kydlíček. Math-Verify: Math verification library. Software repository, 2025. URL https://github.com/huggingface/Math-Verify.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 33:3776–3786, 2025. doi: 10.1109/TASLPRO.2025.3606231.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, 2019.

Neel Nanda. Attribution patching: Activation patching at industrial scale. Mechanistic Interpretability Blog, 2023. URL https://www.neelnanda.io/mechanistic-interpretability/attribution-patching.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. In *Advances in Neural Information Processing Systems*, 2021.

Siyuan Qiao, Zhe Lin, Jianming Zhang, and Alan L. Yuille. Neural rejuvenation: Improving deep network training by enhancing computational resource utilization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 61–71, 2019.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Qwen Team. Qwen3.5: Towards native multimodal agents. Qwen blog, 2026. URL https://qwen.ai/blog?id=qwen3.5.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. Distilling reasoning capabilities into smaller language models. In *Findings of the Association for Computational Linguistics: ACL*, 2023.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL https://www.jmlr.org/papers/v15/srivastava14a.html.

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

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, 2017.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations*, 2023.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022.

Haoyun Xu, Runzhe Zhan, Yingpeng Ma, Derek F. Wong, and Lidia S. Chao. Let's focus on neuron: Neuron-level supervised fine-tuning for large language model. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 9393–9406, 2025.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. MetaMath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations*, 2024.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024.

Andrea Zunino, Sarah Adel Bargal, Pietro Morerio, Jianming Zhang, Stan Sclaroff, and Vittorio Murino. Excitation dropout: Encouraging plasticity in deep neural networks. *International Journal of Computer Vision*, 129:1139–1152, 2021.

A Additional Diagnostic Details

Hot heads and generation coherence. On the 28 high-risk Intermediate Algebra samples at three checkpoints (84 sample/checkpoint units), only 8/84 hot-suppressed and 18/84 hot-amplified generations terminate normally, with mean 4-gram repetition of 0.58 and 0.75, compared with 78/84 and 0.11 for the same-count random-head control.

Sample-level MLP exposure. Among Intermediate Algebra examples that are correct at a given epoch, high exposure to train-hot MLP-C predicts a larger drop in correctness at the next epoch than low exposure (31.3% vs 23.2%, $p = 0.0364$), with the strongest signal in early layers (layer 7: 36.6% vs 17.9%, $p = 0.0002$).

Table 4: Residual causal probe per subject and checkpoint (pooled in Table 1b). The train-hot residual-C dimensions (top 2%, 51 of 2560 per layer) or the same number of random dimensions are scaled at every decoder layer's output during generation.

				Suppress 0.1×		**Amplify 2.0×**	
Subject	**Checkpoint**	n	**Baseline**	**Hot**	**Random**	**Hot**	**Random**
Precalculus	best (4)	75	34.7%	**0.0%**	**24.0%**	**0.0%**	**25.3%**
Precalculus	last (15)	75	36.0%	**0.0%**	**22.7%**	**0.0%**	**28.0%**
Int. Algebra	best (1)	130	34.6%	**0.0%**	**26.9%**	**0.0%**	**26.9%**
Int. Algebra	epoch 8	130	27.7%	**0.0%**	**19.2%**	**0.0%**	**14.6%**

Cohorts and scope. The 75/130 held-out examples here are validation-sized diagnostic cohorts, not the official 546/903-example test sets used later. The 410 MLP probe units comprise 150 Precalculus and 260 Intermediate Algebra sample/checkpoint pairs; repeated measurements of the same problem are not independent examples. Probe baselines are regenerated with the probe's own generation cap and may therefore differ slightly from the training-time held-out evaluation (e.g., 45 vs 47 of 130 at epoch 1). The diagnostic probes use nearest-integer rounding (195 of 9728 MLP neurons; 51 of 2560 residual dimensions) rather than the floor rounding of the training rule (194 MLP neurons).

What these diagnostics establish. The interventions demonstrate component-specific sensitivity of fixed models; the selective new-hot rescue supports a harmful contribution on the tested high-risk subset. They do not identify a universal natural-load threshold, prove that all concentration causes overfitting, or equate mathematical errors with hallucination. The 3.0× point of the dose curve uses fewer pairs than the other points and should not be interpreted as a fully matched five-dose experiment. The epoch correlation and layer-wise transition tests are exploratory; time dependence, repeated examples, sample difficulty, and multiple comparisons limit causal interpretation. These distinctions motivate testing a training intervention rather than assuming that suppressing every hot unit at inference is beneficial.

B Method Details

B.1 Layer Selection

Table 7 compares masking all layers with masking a random 25% or 50% of layers per update, using M on GSM8K. All-layer masking is clearly best for both hybrid models and within one point of the best random-layer variant for the `Qwen3` models, so all other experiments mask every layer. In this ablation, we sample $\lceil qL \rceil$ layers without replacement per optimizer update, with $q \in \{0.25, 0.5, 1\}$. This gives 7/14/28 layers for the 28-layer models and 6/12/24 for the 24-layer models. Component masks may interact serially; combining components is not equivalent to one stronger mask on a shared set of neurons.

What the method encourages, and what it does not guarantee. When a hot unit is unavailable, the remaining computation must support the gold target. This creates an opportunity for alternative

Table 5: The three load channels and the evidence for each component. $C = \mathbb{E}_t|a\, g|$ combines activation and gradient information; darker cells mark stronger support.

Channel	Meaning	Heads	MLP	Residual
A	Activation strength	flat/mixed	mostly flat	retention signal
G	Learning pressure	retention signal	growth signal	retention signal
C	Attribution (load)	uneven and causal	growth, causal, rescue	causal, best predictor

Table 6: Main diagnostic settings. The MLP and residual causal probes pool the best and a later checkpoint of each subject (410 sample/checkpoint units). Causal probes do not update checkpoint weights; they only intervene during generation.

Use	Subject	Held-out size	Checkpoints
Causal MLP/residual probes	Precalculus	75	best epoch 4, last epoch 15
Causal MLP/residual probes	Intermediate Algebra	130	best epoch 1, epoch 8
MLP trajectory	Intermediate Algebra	130	epochs 1–15
Head causal probes	Intermediate Algebra	28 high-risk samples	best epoch 1, epoch 8, epoch 15
New-hot rescue	Intermediate Algebra	90 high-risk sample/checkpoint pairs	epochs 8, 10, 13, 15

Table 7: Layer-selection ablation for M on GSM8K. Test accuracy (%); parentheses show signed percentage-point differences from the matched SFT baseline. Every cell uses the same scorer and model-specific generation budget as its SFT baseline. Top-2% pools throughout; all models use full fine-tuning.

Masked layers	`Q3-0.6B`	`Q3-1.7B`	`Q3.5-0.8B`	`Q3.5-2B`
SFT	47.61	79.38	46.25	63.08
All layers	63.00 (+15.39)	81.12 (+1.74)	55.88 (+9.63)	77.33 (+14.25)
Random 25% of layers	61.71 (+14.10)	81.35 (+1.97)	51.10 (+4.85)	62.40 (-0.68)
Random 50% of layers	63.76 (+16.15)	79.91 (+0.53)	48.67 (+2.43)	60.12 (-2.96)

features to learn. However, the method does not assign a hot unit's knowledge to particular receivers, constrain the final C distribution to be flatter, or guarantee that the unmasked model will use different units. Those are mechanistic hypotheses, examined separately in Section 4.5.

B.2 Pseudocode and Intervention Sites

For the hybrid `Qwen3.5` backbone, whose 24 layers repeat three linear-attention layers and one full-attention layer, the $\mathbf{H}$ mask follows each layer's own output-head layout.

C Training Protocol Details

C.1 Optimization and Selection

Both SFT and PathBalance use AdamW, learning rate 2×10^{-5}, weight decay 0.01, gradient-norm cap 1.0, linear scheduling with 3% warmup, and seed 1333. The schedule horizon is the configured maximum training length, not the eventual early-stop step. The implementation loads BF16 parameters and uses the same optimizer/precision policy in both conditions; it does not introduce FP32 master weights. Gradient checkpointing is non-reentrant. Dynamic padding uses the longest

Algorithm 1 One PATHBALANCE training update. SFT omits the measurement and masking steps and optimizes the same token-normalized CE on the unmasked model.

Require: model f_θ, accumulation window B_s with T_s supervised tokens, target components $\mathcal{K} \subseteq \{\mathrm{M}, \mathrm{R}, \mathrm{H}\}$, hot fraction $\rho = 2\%$

Reset load statistics; remove any intervention hooks
for each microbatch $b \in B_s$ **do** ▷ clean measurement pass, no update
    Forward and backward gold CE on the unmasked model
    Accumulate A, G, C (Eq. 1) over all gold-prediction positions
end for
Discard the parameter gradients of the measurement pass
for each component $k \in \mathcal{K}$ and layer ℓ **do**
    $\mathcal{S}^k_{\ell,s} \leftarrow$ top-max$(1, \lfloor \rho\, d_k \rfloor)$ units by current C ▷ 12.5% for heads
    Install a binary mask $m^k_{\ell,s}$ that zeros $\mathcal{S}^k_{\ell,s}$ (no rescaling)
end for
for each microbatch $b \in B_s$ **do** ▷ masked training pass
    $\mathcal{L}_b \leftarrow$ masked gold CE (Eq. 3); backpropagate $\frac{T_b}{T_s}\mathcal{L}_b$
end for
Apply one AdamW update; remove all masks
if step s is a multiple of 60 **then**
    Generate unmasked validation answers; keep the checkpoint if accuracy strictly improves
end if

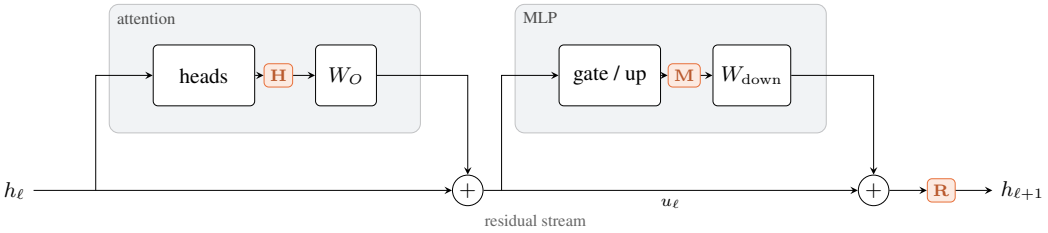


Figure 4: Intervention sites in a decoder layer. **H** masks complete output-head vectors before the attention output projection, **M** masks MLP channels before the down projection, and **R** masks coordinates of the full layer output, including carried information. For `Qwen3.5`, the H mask follows each layer's attention type.

sequence in a microbatch, rounded to a multiple of eight. Effective batches are 32 for GSM8K and 8 for MATH, with a possible smaller last accumulation window.

Generated validation accuracy is evaluated after every 60 optimizer steps. The best checkpoint changes only when the new accuracy is *strictly* higher; ties consume patience. Training stops after ten checks without improvement, not ten epochs. A final partial interval can be evaluated at the configured end of training. Resumable state is saved at validation checkpoints; interrupted runs resume from the latest committed state, which need not be the best checkpoint. Test generation loads the best checkpoint.

C.2 UNMASKED GENERATION AND SCORING

Validation and test both use greedy vLLM generation (Kwon et al., 2023), with identical prompt construction and scoring within each dataset. Training hooks and masks are absent. On large-memory GPUs the trainer and persistent vLLM worker coexist on one GPU without CPU offloading; the 40-GB deployment uses separate training and generation GPUs. Wall-clock times are therefore not directly comparable across hardware.

For GSM8K, the fixed rule extracts a number after the last `####`, otherwise from the last valid box, otherwise the last numeric substring. Decimal-normalized equality determines correctness. This fallback can credit an answer in unfinished reasoning; termination and truncation are therefore reported separately. For MATH, the final outer `\boxed` or `\fbox` answer is compared with the

Table 8: Text-backbone dimensions and training hot-pool sizes. MLP and residual entries are width followed by $\lfloor 0.02d \rfloor$ selected units. For hybrid models, full/linear attention have distinct output-head counts.

Model	Layers	MLP width / hot	Residual width / hot	Heads / hot
`Qwen3-0.6B`	28	3072 / 61	1024 / 20	16 / 2
`Qwen3-1.7B`	28	6144 / 122	2048 / 40	16 / 2
`Qwen3.5-0.8B`	24	3584 / 71	1024 / 20	8 / 1; 16 / 2
`Qwen3.5-2B`	24	6144 / 122	2048 / 40	8 / 1; 16 / 2
`Qwen3-4B`	36	9728 / 194	2560 / 51	32 / 4

Table 9: Training protocol. Budgets and batches are matched between SFT and PATHBALANCE within a model/dataset comparison. GSM8K generation budgets differ across model families and are not pooled as a single identical-budget cross-family experiment.

Setting	GSM8K	Intermediate Algebra	Precalculus
Train / validation / test	6725 / 748 / 1319	1165 / 130 / 903	671 / 75 / 546
Maximum train sequence	768	2816	2816
Maximum evaluation prompt	256	1024	1024
New tokens: `Qwen3`	1024	4096	4096
New tokens: `Qwen3.5`	1536	4096	4096
vLLM context: `Qwen3`	1536	5120	5120
vLLM context: `Qwen3.5`	2048	5120	5120
Microbatch $\times$ accumulation	16×2	4×2	4×2
Optimizer steps per epoch	211	146	84
Maximum epochs	50	25	25
Validation / logging interval (steps)	60 / 30	60 / 30	60 / 30
Patience (validation checks)	10	10	10
Validation / test batch	8 / 8	8 / 8	8 / 8

gold final box using `math-verify` (Kydlíček, 2025). Missing or unparseable predictions count as wrong. There is no last-number fallback for symbolic MATH answers. A failure to parse the gold target aborts scoring rather than silently reducing accuracy. Exact prompt text and scorer identifiers appear in Appendix F.

Reporting scope. All settings are reported individually. Checkpoint selection is validation-only. We do not present the maximum over many settings as an unbiased estimate of a separately selected algorithm. Paired test-example uncertainty is conditional on one training seed and cannot establish run-to-run robustness.

D ADDITIONAL RESULTS

D.1 TRAINING DYNAMICS

In several settings, PATHBALANCE reaches its best validation accuracy at an early check; for example, on GSM8K the `Qwen3-0.6B` M checkpoint is selected at step 60, while SFT selects step 720. Selected steps vary across models and settings, so an early peak is not universal. The ten-check patience rule permits 600 further optimizer steps without improvement. The validation records support a higher best-achieved score in many settings, not a claim of uniformly slower overfitting.

D.2 ACCURACY, TERMINATION, AND ANSWER FORMAT

The response audit in Appendix I separates correctness from token-cap hits, missing extracted answers, and repeated text. Table 11 shows that termination is not uniformly improved by the method. On `Qwen3-0.6B` Precalculus, cap hits fall from 190/546 to 53/546 under M, accompanying improved accuracy. On `Qwen3.5-2B` Precalculus, they instead rise from 133/546 to 303/546, while the M accuracy gain is small. On 4B Intermediate Algebra, cap hits change little (140/903 to

150/903) even though accuracy improves substantially. A single explanation based only on fewer truncated outputs therefore cannot account for the results.

A missing box is distinct from a mathematically wrong but well-formed answer; a token-cap hit is distinct from repetition; and a fluent mathematical error is not by itself evidence of confident hallucination. We retain these categories rather than merging all errors into one mechanism. We do not infer improved calibration or hallucination reduction from answer accuracy.

D.3 Computational Cost

The method adds a clean forward/backward measurement of each accumulation window before the masked forward/backward used for optimization. It therefore does not have the cost profile of ordinary one-pass dropout. Load statistics are reduced to per-layer vectors rather than writing all per-token activations to disk, and the graphs are consumed sequentially rather than retaining two full graphs together. Nevertheless, an extra backward pass is a material training cost. There is no intervention overhead at inference.

Table 12 reports measured optimizer-window and validation times for paired GSM8K runs. Rounded median training costs change from 0.6 to 1.4 seconds for `Qwen3-0.6B`, from 1.0 to 2.0 seconds for `Qwen3-1.7B`, from 0.7 to 1.4 seconds for `Qwen3.5-0.8B`, and from 1.0 to 2.2 seconds for `Qwen3.5-2B`. These are observed step costs, not end-to-end speedup claims: total runtime also depends on selected/stopping steps, validation generation, export, and hardware placement.

E Extended Discussion

Two contributions, with different evidence. The diagnostic contribution identifies nonuniform load, component-specific fragility, and selective rescue in fixed models. The training contribution tests an adaptive intervention motivated by those findings. Its success does not require claiming that every hot unit is harmful, and its remaining mechanistic uncertainty does not erase the diagnostic evidence. Conversely, inference-time fragility does not by itself validate every training control.

Specialization versus dependence. The fact that suppressing useful hot units at inference is harmful is compatible with temporarily masking them during training: the latter supplies gold supervision and allows parameters to adapt. Neither experiment implies that healthy computation must be uniform.

Scope of the evidence. The fixed-checkpoint probes measure how sensitive a model is to its hot units; they do not by themselves show that the growth of concentration caused the later decline, and repeated sample/checkpoint units are not independent replications. Attribution load depends on the chosen coordinates and loss, and our results refer to greedy, non-thinking generation with the scoring rules of Appendix F.

Hardware. Runs used different GPU types with fixed effective batch sizes; kernel differences may slightly change numerical trajectories.

F Reproducibility and Exact Protocol

F.1 Model and Dataset Identity

All training runs use full fine-tuning, seed 1333, and thinking disabled. The five models are loaded from their Hugging Face checkpoints:

- `Qwen3-0.6B`: `https://huggingface.co/Qwen/Qwen3-0.6B`
- `Qwen3-1.7B`: `https://huggingface.co/Qwen/Qwen3-1.7B`
- `Qwen3-4B`: `https://huggingface.co/Qwen/Qwen3-4B-Instruct-2507`
- `Qwen3.5-0.8B`: `https://huggingface.co/Qwen/Qwen3.5-0.8B`
- `Qwen3.5-2B`: `https://huggingface.co/Qwen/Qwen3.5-2B`

GSM8K is loaded from `openai/gsm8k`, subset `main`, default revision. MATH is loaded from `DigitalLearningGmbH/MATH-lighteval`, revision `0530c78699ea5e8eb5530600900e1f328b48acad`, filtered by subject. Validation uses 10% of the official training split with seed 42, not part of the official test split.

F.2 Prompts and Token Accounting

Both training and evaluation use the same system prompt within each dataset. The user's message is the dataset question, including its supplied mathematical notation; no options are removed. The assistant target is the full supplied solution. Disabling the model's dedicated thinking mode does not remove the step-by-step reasoning explicitly requested by these prompts.

GSM8K system prompt.

> You are a mathematics expert. Solve the problem step by step, showing clear reasoning, and put your final answer after `####`.

MATH system prompt.

> You are a mathematics expert. Solve the problem step by step, showing clear reasoning, and put your final answer within `\boxed{}`.

Training constructs the complete chat-template sequence and verifies that its prompt prefix equals the generation prompt. Exactly this prefix is masked from CE. The supervision includes reasoning, final answer, and the template's completion suffix. No extraction-success filter excludes examples from training. Prompt caps include the system message and template overhead. Generation budgets count new tokens only; vLLM context covers prompt plus generation. Overlength gold or prompt inputs trigger an error rather than silently changing the task. Dynamic padding concerns training tensors; vLLM schedules variable-length generation requests without right-padding all prompts to the maximum configured context.

F.3 Scoring and Failure Handling

The GSM8K scorer is `gsm8k_hash_boxed_last_numeric_v3`. Its prediction precedence is the last hash-answer region, then the final valid box, then the final signed numeric substring. Commas are normalized, and decimal equality is used. The gold answer must have a valid final hash-marked number. This is an answer-extraction convention, not a proof that the generated reasoning is valid.

The MATH scorer is `final_boxed_v3`. It reads the last outer box while respecting nested braces. A malformed final box is not replaced with an earlier correct-looking box. A bare mention of the box command before a later valid box does not prevent the latter from being read. Symbolic equivalence is checked with `math-verify`; an empty, unparseable, or timed-out prediction is wrong. Invalid gold parsing is a hard error. Gold sentence punctuation has a narrow cleanup fallback that is not applied to arbitrary malformed predictions. The same scorer is used for validation and test.

F.4 Training-State Semantics

The Hugging Face trainer owns optimizer stepping, scheduling, callbacks, evaluation intervals, and checkpoint state. Its customization adds the clean measurement pass and masked CE, preserves full-window token normalization, and restores data/RNG/callback state on resume. The clean measurement's parameter gradients are discarded before the masked pass. The same masks remain active through backward recomputation, then are removed before validation. Parameter counts in the included metadata confirm that all model parameters are trainable, and saved checkpoints contain the full model.

Every validation checkpoint saves the latest resumable state; best-model selection is a separate strictly improving accuracy record. A tie does not reset patience. The final test explicitly loads that selected checkpoint.

F.5 Result Collection

The main results comprise 105 runs (five models, three datasets, and seven settings including SFT), all with full fine-tuning on the full datasets. All tables are generated from the saved per-example test results.

G Diagnostic Protocol

The diagnostic run fine-tunes `Qwen3-4B` for 15 epochs with full fine-tuning, using a microbatch of two and accumulation of four. It saves a checkpoint after every epoch and measured per-sample activation/gradient information; this diagnostic logger averages load within each sample before averaging across samples, and its position mask is offset by one token from the training-time measurement of Section 4.1. Precalculus uses best epoch 4 and final epoch 15 in the paired MLP probes; Intermediate Algebra uses best epoch 1 and epoch 8, with all 15 checkpoints available for trajectory analysis. These per-epoch checkpoints are distinct from the 60-step validation checks of Section 4.3. The causal generation probes use a 2048-token cap; the residual probe uses 3072 and reproduces the same baseline (134/410).

Aggregation units. The per-example top-mass values (Table 1a and Figure 1, panel 1) and the sample-conditional retention analysis use different aggregation units. They should not be collapsed into a single numerical effect size. A low MLP retention association in one analysis and a strong MLP intervention in another are not logically contradictory: one predicts natural transitions, whereas the other perturbs a fixed model.

The diagnostic head probes target 3/32 heads (9.375%), unlike the 12.5% head pool used in training. The rounded 195-neuron diagnostic MLP intervention also differs by one neuron from the floor-rounded 194-neuron 4B training intervention. Same-count random controls in the diagnostic probes are therefore interpreted within their respective experiment, not reused as training-dropout controls.

Exposure and statistical units. Train-defined hot sets and sample-level exposure are distinct from ranking the responses themselves into high/low exposure groups. An example can have high exposure in some layers and low exposure in others; an averaged exposure statistic does not assert that every layer follows the same route. The 410 MLP, 84 head, and 90 rescue units include sample/checkpoint pairs. Reported significance should be read with that repeated-measure structure and exploratory layer selection in mind. Confidence- or difficulty-adjusted transition analyses are not established by restricting to currently correct examples alone.

H Paired Uncertainty for Training Results

For each M comparison, align test examples by identifier and count SFT-wrong/PB-correct rescues b and SFT-correct/PB-wrong harms c. The paired effect is $100(b-c)/n$ percentage points. The exact two-sided discordant-pair test uses a binomial null with probability one half conditional on $b+c$. Percentile intervals resample test examples 20,000 times, with a fixed analysis seed, retaining pairing. This is equivalent to multinomial resampling of rescued, harmed, and unchanged outcomes. It does not resample training runs or correct for exploratory configuration selection. All reported p values are unadjusted.

I Output and Efficiency Audits

Token-cap hits use vLLM's recorded `finish_reason=length`; missing extraction means the saved prediction field is empty, not every symbolic verification failure. Repetition is $1 - |\text{unique whitespace 4grams}|/|\text{whitespace 4grams}|$, defined as zero for fewer than four words and then averaged over outputs.

Table 10: Paired comparison of the M setting against SFT. Intervals are percentile paired example-bootstrap intervals (20,000 replicates), conditional on these trained checkpoints, not training-seed uncertainty. Exact discordant-pair tests are unadjusted and descriptive.

Model	Dataset	Rescued	Harmed	Δ [95% CI]	p
Q3-0.6B	GSM8K	320	117	+15.39 [12.36, 18.35]	$< 10^{-4}$
Q3-0.6B	Int. Alg.	224	28	+21.71 [18.60, 24.92]	$< 10^{-4}$
Q3-0.6B	Precalc.	123	18	+19.23 [15.20, 23.26]	$< 10^{-4}$
Q3-1.7B	GSM8K	118	95	+1.74 [-0.45, 3.94]	0.1315
Q3-1.7B	Int. Alg.	367	14	+39.09 [35.66, 42.52]	$< 10^{-4}$
Q3-1.7B	Precalc.	203	19	+33.70 [29.12, 38.28]	$< 10^{-4}$
Q3.5-0.8B	GSM8K	293	166	+9.63 [6.44, 12.81]	$< 10^{-4}$
Q3.5-0.8B	Int. Alg.	84	47	+4.10 [1.66, 6.64]	0.0016
Q3.5-0.8B	Precalc.	49	34	+2.75 [-0.55, 6.04]	0.1239
Q3.5-2B	GSM8K	283	95	+14.25 [11.52, 16.98]	$< 10^{-4}$
Q3.5-2B	Int. Alg.	190	52	+15.28 [12.07, 18.60]	$< 10^{-4}$
Q3.5-2B	Precalc.	51	44	+1.28 [-2.20, 4.76]	0.5384
Q3-4B	GSM8K	28	19	+0.68 [-0.30, 1.67]	0.2430
Q3-4B	Int. Alg.	467	14	+50.17 [46.73, 53.60]	$< 10^{-4}$
Q3-4B	Precalc.	283	14	+49.27 [44.69, 53.85]	$< 10^{-4}$

Table 11: Output behavior of matched SFT and M models. Each cell gives SFT / M. Truncation and missing extracted answers are counts, and repetition is the mean repeated whitespace 4-gram fraction. Missing extraction does not include all verifier failures. These descriptive metrics do not establish hallucination.

Model	Dataset (n)	Truncated	No answer	Repetition	Mean tokens
Q3-0.6B	GSM8K (1319)	29 / 7	1 / 0	0.06 / 0.05	145 / 240
Q3-0.6B	Int. Alg. (903)	165 / 103	190 / 99	0.22 / 0.23	990 / 1361
Q3-0.6B	Precalc. (546)	190 / 53	201 / 53	0.38 / 0.22	1567 / 1156
Q3-1.7B	GSM8K (1319)	9 / 1	0 / 0	0.07 / 0.04	273 / 265
Q3-1.7B	Int. Alg. (903)	150 / 92	158 / 92	0.20 / 0.20	969 / 1423
Q3-1.7B	Precalc. (546)	42 / 42	53 / 41	0.14 / 0.17	677 / 1204
Q3.5-0.8B	GSM8K (1319)	0 / 69	0 / 0	0.02 / 0.07	123 / 385
Q3.5-0.8B	Int. Alg. (903)	26 / 413	42 / 417	0.09 / 0.28	434 / 2269
Q3.5-0.8B	Precalc. (546)	35 / 131	65 / 154	0.14 / 0.27	601 / 1313
Q3.5-2B	GSM8K (1319)	17 / 54	0 / 0	0.05 / 0.06	142 / 412
Q3.5-2B	Int. Alg. (903)	204 / 497	210 / 494	0.24 / 0.18	1134 / 2771
Q3.5-2B	Precalc. (546)	133 / 303	125 / 302	0.26 / 0.38	1249 / 2526
Q3-4B	GSM8K (1319)	29 / 21	0 / 0	0.04 / 0.03	294 / 265
Q3-4B	Int. Alg. (903)	140 / 150	154 / 130	0.17 / 0.13	894 / 1807
Q3-4B	Precalc. (546)	44 / 70	50 / 60	0.14 / 0.11	659 / 1583

Table 12: Measured GSM8K optimizer-window cost for SFT / M. Median seconds include the clean measurement pass where present and exclude validation; validation is reported separately. Peak allocated memory is from the trainer process, not total device use including vLLM. These paired runs used the same H100 deployment type; timings are descriptive, not a controlled hardware benchmark.

Model	Train seconds/update	Seconds/validation	Peak allocated GiB
Q3-0.6B	0.6 / 1.4	66.7 / 69.1	21.0 / 22.5
Q3-1.7B	1.0 / 2.0	62.8 / 90.8	29.9 / 32.9
Q3.5-0.8B	0.7 / 1.4	47.3 / 43.0	32.2 / 33.7
Q3.5-2B	1.0 / 2.2	72.5 / 70.0	41.1 / 43.6

I.1 Illustrative Failure and Rescue Cases

These two examples are the first sample IDs, in each subject, for which the `Qwen3-0.6B` SFT output is wrong and truncated while the M output is correct. They illustrate a failure mode, not its frequency.

Intermediate Algebra, sample 7. For the minimum distance from the origin to $x^2+8xy+7y^2 = 225$, the gold answer is 5. SFT consumes 4096 tokens and repeatedly applies AM–GM with escalating exponents, including powers 6291456 and 12582912, without finishing. The M model finishes in 1374 tokens; its final calculation uses $(x, y) = (\sqrt{5}, 2\sqrt{5})$ and returns $\boxed{5}$.

Precalculus, sample 12. For spherical coordinates $(12, 2\pi/3, \pi/4)$, the target is $x + z = 3\sqrt{2}$. SFT reaches 4096 tokens while repeating $\rho\cos\theta\sin\phi+\rho\cos\phi$ without a final answer. The M model finishes in 569 tokens, computes $x = -3\sqrt{2}$ and $z = 6\sqrt{2}$, and returns $\boxed{3\sqrt{2}}$. Neither example demonstrates where alternative computation was learned; the aggregate audit also contains settings in which truncation increases.