SCALING CHANGES THE DIRECTION OF CORRECTNESS CHURN MORE THAN ITS AMOUNT

Anonymous authors
Paper under double-blind review

ABSTRACT

Scaling improves aggregate language model performance, but individual predictions do not improve monotonically. Larger models correct some errors made by smaller models and also turn some previously correct predictions into errors. We characterize these changes using two quantities. Correctness churn, C, measures the fraction of positions whose correctness changes. Directional imbalance, D, measures whether those changes favor the larger model. Across three released model families, their base pretrained checkpoints, three corpora, and six data controlled Pythia transitions from 160M to 12B parameters, 3.1–5.2% of correct next token predictions become incorrect after scaling, with no consistent decrease as model size grows. A similar pattern appears in log likelihood. Positive and negative shifts of 0.289 and 0.275 nats yield a net improvement of only 0.014 nats. PolyPythias provides nine independent runs at each of five model sizes, allowing scaling transitions to be compared with retraining at a fixed size. Reinitialization and data reshuffling reproduce most of the observed churn but produce little directional imbalance. At the 160M$\rightarrow$410M transition, correctness churn is 11.3% under scaling and 9.5% under retraining, with a bootstrap 95% confidence interval of $[+1.45, +2.13]$ percentage points for the difference. Directional imbalance is $D = +0.273$ under scaling and $|D| = 0.024$ under retraining. The directional gap remains positive at every model size after Holm–Bonferroni correction, while larger models increasingly make the same errors. These directional differences also support selective routing. We derive a total variation bound on the maximum gain obtainable from a candidate routing statistic. Peer model disagreement provides an effective signal for separating rescued and broken predictions. A consensus based cascade that escalates positions with peer disagreement improves over universal large model escalation across all evaluated transitions while using less compute, and approaches the empirical ceiling for the routing rules considered.

1 INTRODUCTION

Language model scaling is usually evaluated using aggregate metrics such as perplexity, benchmark accuracy, and scaling curves. These metrics summarize average performance but do not show how individual predictions change as model size increases. When models share a tokenizer, predictions can be compared at the same sequence positions across scales. This comparison reveals substantial movement in both directions. Larger models correct some errors made by smaller models, but they also make errors at positions where smaller models were correct. The net improvement reported by an aggregate metric therefore reflects the difference between two opposing flows.

The size of these flows is substantial relative to the resulting gain. Across three released model families and six scaling transitions, larger models rescue 5.2–7.6% of positions while breaking 4.1–5.0%. A similar pattern appears in log likelihood. For Qwen2.5 3B$\rightarrow$7B, positive and negative shifts of 0.289 and 0.275 nats leave a net improvement of only 0.014 nats. The negative flow persists in base pretrained checkpoints, on additional corpora, and across a data controlled Pythia ladder spanning 160M to 12B parameters. Bidirectional changes also appear on ARC-Challenge, HellaSwag, and LAMBADA.

Previous work has documented prediction regressions in finetuned models (Zhong et al., 2021) and across consecutive model deployments (Milani Fard et al., 2016). We study the same phenomenon

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.

under parameter scaling and separate two properties that are combined in aggregate accuracy. Let R denote the fraction of positions rescued by the larger model and B the fraction broken by it. We define correctness churn as

$$C = R + B,$$

which measures how often correctness changes, and directional imbalance as

$$D = \frac{R - B}{R + B},$$

which measures whether those changes favor the larger model. The change in top-1 accuracy satisfies

$$\Delta\mathrm{Acc} = CD.$$

Models can therefore exhibit similar amounts of churn while differing substantially in the direction of those changes.

A direct comparison across released model families cannot isolate the effect of scale because model variants may also differ in data mixture, optimization settings, and training procedures. We first examine Pythia, where all model sizes are trained on the same data in the same sequence order. The same bidirectional pattern remains across six size transitions. We then use PolyPythias, which provides nine independent pretraining runs at each of five model sizes with differences in random initialization and data ordering. Independent runs at the same size provide a baseline for the amount of prediction change produced by training stochasticity alone.

The replicated comparison shows that training stochasticity accounts for much of the amount of correctness churn, while scaling has a much larger effect on its direction. At the 160M→410M transition, scaling produces $C = 11.28\%$, compared with 9.52% between independent 160M runs. The difference is $+1.76$ percentage points with a bootstrap 95% confidence interval of $[+1.45, +2.13]$. Directional imbalance differs more strongly. Scaling gives $D = +0.273$, while same-size retraining gives $|D| = 0.024$. The same pattern holds across all four PolyPythias size transitions. The difference in churn remains between $+0.9$ and $+3.2$ percentage points, while the directional gap remains positive after Holm–Bonferroni correction. Larger models also increasingly agree on which errors they make.

This distinction is relevant to selective inference. A router can improve over always using the larger model only if it can identify positions where scaling is more likely to repair an error than to break a correct prediction. For a statistic s available to a routing policy, we derive a total variation bound on the maximum accuracy gain achievable by any decision rule based on s. The bound allows candidate routing signals to be evaluated before fitting a router. Model-internal confidence statistics provide little separation between rescued and broken positions. Disagreement with peer models provides substantially more separation.

Peer disagreement identifies positions where predictions remain unstable across independently trained or differently sized models. In the PolyPythias runs, the probability that a correct prediction is broken by the larger model is 1.5% under unanimous peer agreement and rises to 49.9% under majority disagreement. The same signal also identifies errors that are more likely to be repaired. In released model families, a consensus based cascade escalates only positions selected by peer disagreement. For Qwen2.5 3B→7B, escalating the 31% of positions where Qwen2.5-0.5B disagrees with Qwen2.5-3B yields a gain of $+1.23$ points, compared with $+0.87$ points from using 7B at every position, while requiring less compute.

Our main contributions are:

- We decompose prediction changes under scaling into rescue and break flows, and separate the amount of correctness churn C from its directional imbalance D. Across released models, pretrained checkpoints, multiple corpora, and a data controlled scaling ladder, the negative flow remains substantial as model size increases.
- We use replicated pretraining runs to compare scaling with training stochasticity at fixed model size. Training randomness reproduces most of the churn volume, whereas increasing model size produces substantially stronger directional alignment across all PolyPythias transitions.
- We connect the directional structure of prediction changes to selective routing. A total variation bound characterizes the maximum gain available from a routing statistic, and peer model disagreement provides a practical signal for a consensus based cascade that improves over universal large model escalation in the evaluated transitions.

2 Related Work

Instance-level changes and prediction churn. Prior work has examined whether larger or updated models improve uniformly at the instance level. Zhong et al. (2021) show that larger BERT variants can underperform smaller ones on 1–4% of examples after finetuning. Their analysis varies finetuning seeds while holding pretrained weights fixed, whereas we study next-token prediction and vary pretraining runs directly. Related work on model updates studies prediction churn (Milani Fard et al., 2016), negative flips, and backward-compatible updates (Xie et al., 2021; Yan et al., 2021; Träuble et al., 2021; Lai et al., 2023), with mitigation methods based on label smoothing (Bahri & Jiang, 2021) and distillation (Jiang et al., 2022). Regressions have also been reported for LLM updates (Echterhoff et al., 2024). We distinguish total disagreement Q from correctness churn $C = R + B$ and analyze positive and negative flows jointly across parameter scales.

Routing and scaling efficacy. Small-model uncertainty has been used for speculative decoding (Leviathan et al., 2023), model cascades (Gupta et al., 2024), and learned routing (Ong et al., 2025). Narayan et al. (2022) show that larger models help more where smaller models are uncertain. We separate these gains into rescue and break distributions and find that both concentrate in similar uncertainty regions. Routing therefore depends on separating the two flows rather than identifying uncertainty alone. Post-hoc temperature scaling does not change the top-1 flows because it preserves the argmax prediction (Guo et al., 2017).

Training dynamics and run-to-run variation. Individual examples can be learned and forgotten during training (Toneva et al., 2019), differ in acquisition timing (Paul et al., 2021; Baldock et al., 2021), and interact through gradient interference (Mircea et al., 2025). Variation across random seeds is also well documented (Picard, 2021). Pythia provides a size-controlled ladder with fixed training data and sequence order (Biderman et al., 2023), while PolyPythias provides multiple independent pretraining runs across model sizes (van der Wal et al., 2025). We use these runs to compare scaling-induced prediction changes with those arising from training stochasticity at a fixed model size.

3 Measuring Signed Prediction Flows

Aligned positions. Within each architectural family, models share a tokenizer, enabling exact position-by-position comparisons across scales. We analyze three open families from two providers: Qwen3 (4/8/14B), Qwen2.5 (0.5/1.5/3/7B), and Gemma-3 (4/12B) (Yang et al., 2025; 2024; Gemma Team et al., 2025), using instruction-tuned checkpoints to reflect serving deployments; base pretrained models display qualitatively identical flows (Appendix A.3). Across 120 shared documents, we log the true-token log-probability, sequence entropy, and greedy prediction at every position ($\approx 59{,}000$ positions per model). The corpus spans 70 multi-session dialogues (LoCoMo), 31 multi-hop QA contexts (HotpotQA via LongBench), and 19 extended chat sessions (LongMemEval (Maharana et al., 2024; Yang et al., 2018; Bai et al., 2024; Wu et al., 2025)), each truncated to 512 tokens to fix computational budgets and avoid context-length confounds. See Appendix A.1 for sampling procedures, domain breakdowns, and 1024-token checks.

The contingency table. Let S and L denote the smaller and larger models, $\hat{y}_S, \hat{y}_L$ their greedy outputs, and $C_S, C_L \in \{0, 1\}$ indicator variables for correctness against target y. We partition the joint prediction space into three disjoint outcomes:

$$R = P(C_S{=}0, C_L{=}1), \quad B = P(C_S{=}1, C_L{=}0), \quad W = P(C_S{=}C_L{=}0,\ \hat{y}_S \neq \hat{y}_L), \tag{1}$$

where R is the *rescue* rate and B is the *break* rate. This formulation distinguishes total prediction disagreement from accuracy-altering changes:

$$Q = R + B + W = P(\hat{y}_S \neq \hat{y}_L), \qquad C = R + B, \tag{2}$$

where Q denotes total *prediction churn* (Milani Fard et al., 2016) and C isolates *correctness churn*. This distinction is non-trivial: on our suite, $Q = 28.8\%$ while $C = 11.3\%$, showing that under two-fifths of disagreements impact top-1 performance. Net accuracy change depends entirely on C and its directional orientation:

$$\Delta\text{Acc} = R - B = C \cdot D, \qquad D = \frac{R - B}{R + B} \in [-1, 1], \tag{3}$$

Table 1: Every size step we measure. Top-1 is a two-by-two contingency so its net is exactly $R - B$; likelihood is not and needs its own gross flows. $r_{\text{fix}} = P(L \text{ right} \mid S \text{ wrong})$ and $r_{\text{reg}} = P(L \text{ wrong} \mid S \text{ right})$ condition on the pool each flow draws from, and $P(d{<}0)$ is the share of positions where the larger model assigns lower probability to the true token.

		top-1 (%)			conditional (%)		log-likelihood (nats)			
family	step	R	B	net	r_{fix}	r_{reg}	G	$\mathcal{L}$	net	$P(d{<}0)$
Qwen3	4B→8B (2.0×)	5.29	**4.08**	+1.21	11.2	**7.7**	0.419	**0.220**	+0.199	47%
	8B→14B (1.8×)	5.15	**4.49**	+0.66	11.2	**8.3**	0.321	**0.245**	+0.075	51%
Gemma-3	4B→12B (2.8×)	6.66	**4.89**	+1.76	14.3	**9.1**	0.573	**0.280**	+0.293	50%
Qwen2.5	0.5B→1.5B (3.1×)	7.61	**4.91**	+2.70	15.3	**9.8**	0.430	**0.207**	+0.223	36%
	1.5B→3B (2.0×)	5.86	**4.74**	+1.12	12.4	**9.0**	0.280	**0.232**	+0.048	42%
	3B→7B (2.5×)	5.90	**5.03**	+0.87	12.8	**9.3**	0.289	**0.275**	+0.014	41%

where D denotes the *normalised directional imbalance*. $D = 1$ indicates pure gain, $D = 0$ perfect cancellation, and the fraction of positive flips is $(1 + D)/2$ (e.g., $D = +0.273$ corresponds to 63.7% rescues). Reporting only aggregate ΔAcc conflates the volume of churn C with its directional bias D.

Likelihood decomposition. While top-1 accuracy splits into discrete cells, log-likelihood differences $\Delta\text{LL} = \mathbb{E}[d]$ with $d = \log p_L(y) - \log p_S(y)$ represent scalar net shifts. We unpack these shifts into opposing flows: $G = \mathbb{E}[\max(d, 0)]$ and $\mathcal{L} = \mathbb{E}[\max(-d, 0)]$, yielding $\Delta\text{LL} = G - \mathcal{L}$, likelihood churn $C_{\text{LL}} = G + \mathcal{L}$, and imbalance $D_{\text{LL}} = (G - \mathcal{L})/C_{\text{LL}}$. Evaluating discrete breaks (B) alongside continuous degradation ($\mathcal{L}$) verifies that bidirectional flipping is not an artifact of greedy argmax selection.

Conditioning and bootstrap intervals. Because base accuracy varies across model scales, marginal denominators shift; we therefore also report conditional rates $r_{\text{fix}} = P(C_L{=}1 \mid C_S{=}0)$ and $r_{\text{reg}} = P(C_L{=}0 \mid C_S{=}1)$. To handle token dependencies within documents, confidence intervals for model pairs are computed via document-level bootstrapping ($n = 120$, 4,000 resamples), while training stochasticity is evaluated across independent runs. Routing thresholds are cross-validated on disjoint document splits.

4 Scaling Moves Predictions in Both Directions

Our primary empirical results appear in Table 1, with corresponding flow trajectories illustrated in Fig. 1. Across all evaluated families and rungs, scaling induces substantial bidirectional churn: conventional net gains reflect merely the small residuals surviving widespread cancellation.

Persistence of the negative flow across domains. The break rate remains remarkably constrained, occupying a narrow band of 4.08–5.03% across all family steps—spanning a $16\times$ parameter range—with conditional regression rates r_{reg} consistently within 7.7–9.8%. Although bootstrap intervals confirm the rate is not strictly constant (4.08% $[3.91, 4.24]$ vs. 5.03% $[4.79, 5.27]$), scaling parameter capacity exhibits no monotonic downward effect on regressions. This stability holds across held-out WikiText-103 (4.01–5.22%, Appendix A.4) and pretraining web crawl C4 (Merity et al., 2017; Raffel et al., 2020) (3.72–4.56%, Appendix A.5). Evaluating base pretrained checkpoints reduces breaks by only 0.56 points on average (3.30–4.82%, Appendix A.3), showing that post-training slightly amplifies, rather than introduces, this negative flow.

Shrinking top-1 gains alongside persistent likelihood losses. Along the Qwen2.5 series, rescue rates fall sharply ($7.61\% \to 5.86\% \to 5.90\%$), contracting net gains from $+2.70$ to $+0.87$ points. Diminishing returns under scaling thus stem from fewer newly resolved tokens against an invariant floor of broken ones. A continuous analogue emerges in log-likelihood: gross loss $\mathcal{L}$ remains non-decreasing with scale, and under the strictly controlled regime of Section 5, it increases monotonically across transitions and corpora.

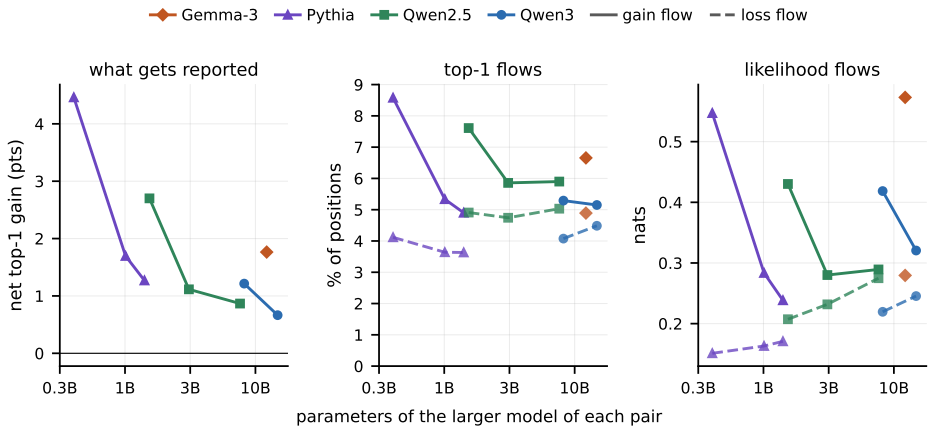

Figure 1: What a scaling curve reports (left) and what it is made of (middle, right), for three released families and the controlled Pythia suite. The net contracts with size while both flows stay large. In the middle panel the gain flow descends steeply and the loss flow does not; in the right panel the gross likelihood loss holds or rises while the gross gain falls, so the shrinking net is not a shrinking loss.

Net improvements versus underlying gross flows. Across 36–51% of positions, the larger model assigns lower probability to the true token than its smaller counterpart. In the narrowest step (Qwen2.5 3B→7B), the net log-likelihood gain across 59,275 positions is just $+0.0145$ nats (bootstrap CI $[+0.0010, +0.0281]$). This marginal net is stable against extreme outliers ($+0.0133$ and $+0.0222$ after 1% and 5% tail trimming) and numerical precision artifacts. Yet, this near-zero net obscures massive underlying probability churn: gross opposing shifts of 0.289 nats (G) and 0.275 nats ($\mathcal{L}$) mean that roughly thirty-nine nats of probability mass are displaced for every single net nat gained.

Churn, persistence, and non-monotone trajectories. Nearly half of the predictions broken at one scaling step recover at the next (43.4% and 48.3% vs. a $\sim$ 12.5% base rate), indicating that much of the regression is optimization churn rather than structural inability. Still, a persistent residue is permanently lost: 11.2% of tokens correctly identified by Qwen2.5-0.5B remain incorrect at 7B. These losses are driven neither by ceiling effects—breaks rarely concentrate where smaller models are highly confident ($p > 0.9$)—nor by calibration differences, as greedy outputs are temperature-invariant. These dynamics remain consistent across genres and at 1024-token context lengths (Appendices A.1, A.7 and B.8).

Extension to benchmark-level evaluations. Applying this decomposition to instance-level task accuracy on ARC-Challenge, HellaSwag, and LAMBADA (Clark et al., 2018; Zellers et al., 2019; Paperno et al., 2016) reveals amplified gross flows (Appendix C.1): scaling rescues 13.1% of instances while breaking 6.6% on average. Strikingly, for Qwen2.5 1.5B→3B on ARC-Challenge, a 9.64% rescue rate is fully offset by a 10.15% break rate, resulting in a net decline of -0.51 points ($[-3.07, +2.13]$) where parameter growth offers zero aggregate benefit.

5 What the Flows are a Property of

Model variants within released families vary simultaneously in pretraining tokens, data mixtures, training recipes, and tokenizers. Consequently, Table 1 characterizes deployed artifacts rather than parameter scaling in isolation. This section systematically controls for these confounding factors.

5.1 Holding the Data Fixed

The Pythia suite trains every model size on the identical corpus in a fixed sequence order (Biderman et al., 2023). While architectural width and depth scale alongside optimizer learning rates (6.0×10^{-4}

at 160M to 1.2×10^{-4} at 12B), data composition and presentation order remain strictly invariant across six scaling steps from 160M to 12B (Appendix B.1).

Across this 75-fold parameter range, the break rate remains virtually flat within a narrow 0.62-point band (3.50%–4.12%), matching trends on WikiText-103 (3.82%–4.08%) and C4 (3.33%–3.64%). Similarly, gross likelihood loss $\mathcal{L}$ exhibits no downward drift, remaining between 0.147 and 0.171 nats. In contrast, all positive quantities diminish: the rescue rate drops from 8.58% to 4.26%, net accuracy gains contract from $+4.46$ to $+0.75$ points, and directional imbalance D decays monotonically from $+0.351$ to $+0.097$ across both corpora (Appendix B.2).

Decomposing the decline in D. Part of the decline in D is mechanically constrained: rescues draw exclusively from positions mispredicted by the smaller model, a candidate pool that shrinks from 55.9% to 46.7% along the ladder. Recomputing D by holding conditional transition rates ($r_{\text{fix}}, r_{\text{reg}}$) fixed to ladder means accounts for 70% of this decline on our corpus and 86% on WikiText-103. The remaining decrease reflects true convergence: r_{fix} falls from 15.3% to 9.1% and r_{reg} from 9.3% to 6.6%. This marginal artifact does not affect Section 5.3, where model pairs are matched in size and accuracy (Appendix B.3).

5.2 More Training Behaves Like More Parameters

Fixing model size at 160M while evaluating intermediate checkpoints (13k, 26k, 52k, 104k, and the final 143k release) isolates the effect of training duration under identical parameters, data, and optimizer states. Across these intervals, the model breaks 3.80% of predictions on average, matching the parameter-scaling break rate to three significant figures. Between step 104k and completion, the model breaks 3.49% of its predictions on our mixture and 3.38% on WikiText-103, despite divergent net shifts (-0.04 vs. $+0.81$ points). Thus, late-stage training churn occurs independently of the sign of the net gain.

5.3 Scaling as a Replicated Experiment

To disentangle parameter scaling from arbitrary optimization divergence, models must share architectures, data sequences, and compute budgets while differing solely in training randomness. PolyPythias provides nine independent pretraining runs across five sizes (14M to 410M) with varied initializations and data shuffling (van der Wal et al., 2025). Pairing runs across adjacent rungs by seed identifier establishes nine replicated scaling transitions, benchmarked against same-size baseline pairs.

Because scaling transitions have an inherent orientation (small to large), their direction D is signed. Conversely, intra-size pairs are exchangeable, rendering signed means arbitrary; we therefore evaluate the magnitude $|D|$ via mean and median to prevent skew from outlier runs (Table 13, Appendix B.4).

Direction separates scale from retraining. At the 160M→410M transition, scaling yields correctness churn $C = 11.28\%$ versus 9.52% for retraining at 160M ($+1.76$ points, bootstrap 95% CI $[+1.45, +2.13]$; $+1.89$ points on WikiText-103). Scaling clearly introduces incremental churn. However, directional imbalance diverges sharply: the size step achieves $D = +0.273$, whereas intra-size retraining yields $|D| = 0.024$ (median 0.023). This $+0.249$ directional gap ($[+0.120, +0.330]$; $+0.280$ on WikiText-103) represents an eleven-fold increase in directional alignment despite only a $1.18\times$ difference in churn volume. Furthermore, prediction disagreement Q far exceeds correctness churn C (28.76% vs. 11.28% for scaling; 26.06% vs. 9.52% for retraining), showing that roughly three-fifths of all prediction flips remain accuracy-neutral across both regimes.

Sensitivity to optimization outliers. At 410M, run-to-run $|D|$ rises to 0.151, driven by a single anomalous run (median $|D| = 0.060$; jackknife means span 0.063–0.169). Conservatively evaluated, the scaling-to-retraining directional ratio at 410M remains between two- and four-fold. Notably, this outlier run fully reverses one scaling transition: while eight seed-matched pairs produce $D \in [+0.239, +0.396]$, pairing with the degraded 410M run drops accuracy by 2.32 points ($D = -0.216$). Optimization variance at scale can thus entirely negate the expected gain of a $2.5\times$ parameter expansion.

Table 2: The replicated comparison repeated at every rung PolyPythias publishes. Each row pairs nine runs at one size with the nine at the next and measures them against the same-size pairs built from the runs on the smaller side, so every row carries its own baseline. Intervals bootstrap over training runs. The direction gap excludes zero at all four rungs and on both corpora, while the amount gap is small and positive throughout: scaling adds churn, and what it mainly does is align it. The last two columns give C/Q, the share of disagreement that changes whether a prediction is correct, which rises monotonically with size on both corpora.

			correctness churn C (%)			direction			C/Q (%)	
size step	×par	n	size	run-run	gap (95% CI)	D	$\lvert D\rvert$	gap (95% CI)	ours	WT103
14M→31M	2.21	9	12.98	10.83	[+1.57,+2.81]	**+0.371**	0.139	**[+0.145,+0.317]**	29.2	21.2
31M→70M	2.26	9	12.90	11.24	[+0.98,+2.33]	**+0.311**	0.143	**[+0.081,+0.254]**	32.6	25.2
70M→160M	2.29	9	13.96	11.10	[+2.36,+3.20]	**+0.452**	0.071	**[+0.345,+0.425]**	37.6	30.8
160M→410M	2.56	9	11.28	9.52	[+1.42,+2.12]	**+0.273**	0.024	**[+0.118,+0.332]**	39.2	33.8

5.4 The Comparison Repeated at Every Rung

Evaluating all four adjacent rungs across PolyPythias (parameter ratios 2.21–2.56) replicates these dynamics, with each transition evaluated against its own intra-size baseline (Table 2).

Across the ladder, the directional advantage of scaling strictly excludes zero at all four rungs across both corpora ($D \in [+0.273, +0.452]$ vs. $|D| \in [0.023, 0.151]$), surviving Holm-Bonferroni correction (Appendix B.5; full 81-pair cross-products yield identical results). The incremental churn amount gap also remains consistently positive ($+0.9$ to $+3.2$ points).

Convergence of joint error representations. The proportion of accuracy-relevant disagreement C/Q increases from 29.2% to 39.2% along the ladder. While partly driven by the shrinking mutual-error pool, the proportion of joint errors where models diverge (ρ) decreases from 0.507 to 0.362 (Appendix B.3). Hence, larger models not only err less frequently, but increasingly make the *same* errors.

In summary, negative prediction flows are not artifacts of shifting data regimes; isolated optimization variance reproduces the vast majority of churn with negligible directional bias. Bidirectional churn is an inherent property of training language models, and scale primarily determines the direction in which those prediction flows align.

6 Detecting the Flows from Inside the Smaller Model

If a smaller model could anticipate which predictions the larger model would repair, regressions could be circumvented by escalating only those positions. We establish a formal framework to evaluate any candidate routing signal prior to training, demonstrate that internal confidence fails this check, and introduce a multi-model consensus signal that succeeds.

6.1 A Certificate for Any Candidate Statistic

Let $s = s(x)$ denote any statistic computed from the smaller model's output. A router corresponds to an escalation decision rule E within the σ-algebra generated by s (subsuming threshold rules $\{s \geq t\}$ and learned transforms). Letting $a = P(R)$, $b = P(B)$, and f_R, f_B represent the conditional densities of s on rescued and broken tokens, escalating on E yields a top-1 accuracy change of $\int_E (af_R - bf_B)$. The optimal rule is thus $E^\star = \{af_R > bf_B\}$, with the maximal attainable gain bounded by:

$$G^\star = \int \max\big(af_R(s) - bf_B(s), 0\big)\, ds = \tfrac{a-b}{2} + \tfrac{1}{2} \int |af_R - bf_B| \;\leq\; (a-b) + a \cdot \mathrm{TV}(f_R, f_B), \quad (4)$$

using $\int |af_R - bf_B| \leq 2a\,\mathrm{TV}(f_R, f_B) + |a - b|$ for $a \geq b$ (Appendix D.1). Because $a - b$ is the baseline return from uniform escalation, the maximum value-add of selective routing on s is strictly bounded by $a \cdot \mathrm{TV}$. This provides a certificate on candidate utility from paired samples without training an explicit router.

Table 3: Routing on peer disagreement, charged what it spends and fitted on a disjoint half of the documents. A signal taking $m + 1$ values offers $m + 2$ policies and no budget dial, so the router is reported at the cost of the policy it uses and the entropy router is given exactly that much compute. On the left the peer is a cheaper rung of the deployed model's own ladder; on the right it is a model of the other family, which is dearer and separates the flows better. Costs are forward-pass units per deployed-model pass.

	always-large		peer from the same ladder					peer from the other family			
step	gain	cost	peer	esc.	cost	entropy	vs always (95% CI)	peer	esc.	cost	vs always
1.5B→3B	+1.12	2.01	0.5B	29%	1.89	+0.96	**+0.23** [+0.04,+0.41]	—	—	—	—
3B→7B	+0.87	2.47	0.5B	31%	1.93	+0.55	**+0.37** [+0.17,+0.57]	Qwen3-4B	28%	2.98	**+0.60**
8B→14B	+0.66	1.80	4B	21%	1.87	+0.67	**+0.18** [+0.01,+0.35]	7B	27%	2.41	**+0.78**

Because empirical plug-in estimates $\widehat{\mathrm{TV}}$ from histograms are positively biased, every reported $\widehat{\mathrm{TV}}$ is evaluated against a permutation null obtained by shuffling rescue and break labels. This yields a conservative baseline for confidence metrics and a stringent test for our proposed signal (Appendix D.1).

6.2 Confidence Does Not Separate the Flows

Ranking positions by the smaller model's surprise at the target token, $-\log p_S(y)$ (the *answer-aware score*), separates the two flows almost completely. Although not deployable without ground-truth labels, it provides an empirical upper bound on achievable separation. In deployed settings, binning by sequence entropy leaves rescue and break densities heavily overlapped across all confidence tiers (Appendix D.3).

Our certificate formalizes this limitation quantitatively (Appendix D.2). Sequence entropy yields $\widehat{\mathrm{TV}}$ between 0.049 and 0.083 against nulls of 0.043–0.052—indistinguishable from an uninformative statistic on one step and clearing the null by only 0.009 and 0.018 on the others. Consequently, $G^\star$ exceeds $R - B$ by at most 0.12 points, whereas the answer-aware score achieves $\mathrm{TV} \in [0.490, 0.578]$, securing 2.5–3.3 points. Equivalent failures hold for top-token probability $p_{\max}$, margin scores, gradient-boosted ensembles on internal features (Appendix D.4), and token-level self-consistency (Wang et al., 2023). Internal confidence measures fail to beat uniform escalation; effective separation requires signals external to the deployed model.

6.3 Separating Flows via Peer Disagreement

Confidence metrics conflate settled errors with unsettled correct predictions. Our control experiments suggest an alternative diagnostic: because retraining produces cancelling churn, inter-model disagreement isolates positions where predictions remain stochastic.

Let S denote the deployed model and $p_1, \ldots, p_m$ denote peer models. We define the consensus rate:

$$\alpha(x) = \frac{1}{m} \sum_{j=1}^{m} \mathbf{1}\Big[\arg\max_v p_j(v \mid x) = \arg\max_v p_S(v \mid x)\Big], \tag{5}$$

escalating positions where α is minimized. This formulation is label-free, and peers can be drawn directly from existing smaller rungs within a deployed serving stack.

Empirical measurements validate this hypothesis. Across nine 160M PolyPythias runs, conditioned on S being correct, break probability $P(\text{break})$ is 1.5% under unanimous peer agreement but rises to 49.9% under majority disagreement. Conditioned on S being incorrect, $P(\text{rescue})$ increases from 3.9% to 24.7%. The lowest-α decile yields $+19.4$ points per escalated token versus $+0.3$ for the highest decile, and $\widehat{\mathrm{TV}}(\alpha)$ reaches 0.102–0.145 against a 0.010 null.

Computational economics of routing. Policies deciding prior to escalation can save compute; post-hoc arbitration cannot. The policy class contains always-large as a fallback, although cross-fitted selection can still incur small held-out deviations (Table 3). Because α admits $m + 1$ discrete

values ($m + 2$ deterministic policies), we evaluate routers at their exact operational expenditure: $s + \sum_j p_j + \phi b$ compute units (non-embedding parameters), compared to $s + \phi b$ for internal routers and b for universal escalation. Utilizing a single cheaper peer, the cascade operates at 1.87–$1.93\times$ the cost of the base model. At matched compute budgets, it achieves accuracy gains of $+1.34$, $+1.23$, and $+0.85$ points (versus $+0.96$, $+0.55$, and $+0.67$ for entropy routing), outperforming universal large-model escalation by $+0.23$, $+0.37$, and $+0.18$ points while reducing compute in two of three configurations.

An *arbiter* executing the larger model globally and retaining answers based on peer consensus buys $+0.26$, $+0.63$, and $+0.30$ points over always-large, but at 3.3–$4.1\times$ cost (Appendix E.2). Cross-family peers achieve comparable margins more efficiently.

Lineage independence in peer selection. Using cross-family peers sharing identical tokenizers (e.g., Qwen2.5 and Qwen3) further sharpens separation. Cross-family peers clear the permutation null by 0.166–0.208 (versus 0.092–0.134 for intra-family peers), improving gains over always-large to $+0.60$–$+0.78$ points (Appendix E.3). For instance, routing Qwen3-8B to 14B with Qwen2.5-7B as a peer yields $+1.45$ points (vs. $+0.66$ for always-large). The cascade is strictly cheaper than universal escalation when $b/s > (1 + \sum_j p_j/s)/(1 - \phi)$, satisfied by two of our three setups (Appendix E.4).

Operating limits defined by D. The cascade outperforms universal escalation if and only if unescalated tokens are net-harmful under the larger model. Rearranging Eq. (4) establishes the operating condition:

$$\max_E \text{ gain} - (R - B) = \int \left(bf_B - af_R\right)_+ ds > 0 \iff \frac{f_B(s)}{f_R(s)} > \frac{1+D}{1-D} \text{ somewhere.} \quad (6)$$

This density threshold rises from 1.15 at $D = 0.069$ to 1.65 at $D = 0.245$. Our token-level steps satisfy this condition, yielding rejected regions worth -0.18 to -0.39 points. On benchmark evaluations, higher directional alignment (average $D = +0.245$) ensures unescalated regions remain net-positive, causing fitted routers to escalate 89% of instances and land within 0.23 points of always-large (Appendix E.1).

Attainment of theoretical limits. Rather than reflecting router deficiency, this boundary matches the empirical ceiling for any rule partitionable over observable states. The cross-fitted ceiling is $+0.23$, $+0.46$, and $+0.18$ points at the token level; our same-ladder cascade reaches the ceiling on two of the three steps and comes within 0.09 points on the third. Across benchmark tasks, the mean cross-fitted ceiling is only $+0.03$ points (Table 24, Appendix E.5).

7 Conclusion

Aggregate improvements from language model scaling can conceal substantial changes in individual predictions. Across the models studied, small net gains arise from much larger rescue and break flows in opposite directions. For Qwen2.5 3B→7B, a log-likelihood gain of 0.014 nats results from positive and negative flows of 0.289 and 0.275 nats. The same pattern persists across base pretrained checkpoints, multiple corpora, and six data controlled Pythia transitions from 160M to 12B parameters. Replicated PolyPythias experiments show that training stochasticity reproduces most of the observed correctness churn, while increasing model size adds only $+0.9$ to $+3.2$ percentage points of additional churn. The larger effect of scaling is on direction, consistently shifting prediction changes toward rescues. This directional structure also governs the value of selective inference. Our total variation analysis bounds the gain available from a routing statistic, and peer model disagreement provides a practical signal for separating positions that are more likely to be rescued from those that are more likely to be broken. The resulting consensus based cascade improves over universal large model escalation in the evaluated transitions while reducing compute in applicable settings. These findings show that the main effect of scaling is not simply how many predictions change, but how those changes are directed.

Reproducibility Statement

All empirical results in this paper are generated via deterministic scripts in our open-source repository from precomputed measurement arrays, with every table and figure explicitly citing its source data file. Across all evaluated models and the 120 evaluation documents, measurement scripts record per-position true-token log-probabilities, entropy values, greedy token predictions, and ground-truth targets. All downstream analyses—including bidirectional decompositions, conditional transition rates, routing curves, and theoretical bounds—recompute from these arrays in minutes on a standard CPU.

Evaluations use public model checkpoints at pinned commit revisions, including all 45 PolyPythias checkpoints (9 independent seeds across 5 model sizes). Statistical intervals are computed via bootstrapping with fixed random seeds, resampling documents for fixed model pairs and pretraining runs for replication axes. Permutation null distributions for total variation estimates share these identical fixed seeds. All routing decision thresholds, peer selections, and classifier parameters are cross-fitted on disjoint document splits with fixed split seeds. Finally, an automated verification script re-derives every claim in the text directly from the saved arrays, asserting exact reproduction.

Use of large language models

ChatGPT and Claude were used to assist with polishing and revising the manuscript and with optimizing experimental code. The initial experimental code was implemented by the authors, and all LLM-assisted code modifications and optimizations were subsequently reviewed and verified by the authors. The authors independently conducted the primary literature review and identified the relevant prior work. After this process, ChatGPT was used only as a supplementary search tool to check for potentially overlooked related studies. The authors take full responsibility for the content of the manuscript, the implementation of the experiments, and the reported results.

References

Dara Bahri and Heinrich Jiang. Locally adaptive label smoothing improves predictive churn. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *PMLR*, pp. 532–542, 2021.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 3119–3137, 2024.

Robert J. N. Baldock, Hartmut Maennel, and Behnam Neyshabur. Deep learning through the lens of example difficulty. In *Advances in Neural Information Processing Systems 34*, 2021.

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 *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430, 2023.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Jessica Maria Echterhoff, Fartash Faghri, Raviteja Vemulapalli, Ting-Yao Hu, Chun-Liang Li, Oncel Tuzel, and Hadi Pouransari. MUSCLE: A model update strategy for compatible LLM evolution. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7320–7332, 2024.

Gemma Team, Aishwarya Kamath, et al. Gemma 3 Technical Report. *arXiv preprint arXiv:2503.19786*, 2025.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330, 2017.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language model cascades: Token-level uncertainty and beyond. In *The Twelfth International Conference on Learning Representations*, 2024.

Heinrich Jiang, Harikrishna Narasimhan, Dara Bahri, Andrew Cotter, and Afshin Rostamizadeh. Churn reduction via distillation. In *International Conference on Learning Representations (ICLR)*, 2022.

Yi-An Lai, Elman Mansimov, Yuqing Xie, and Yi Zhang. Improving prediction backward-compatiblility in NLP model upgrade with gated fusion. In *Findings of the Association for Computational Linguistics: EACL 2023*, pp. 1010–1022, 2023.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286, 2023.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 13851–13870, 2024.

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

Mahdi Milani Fard, Quentin Cormier, Kevin Canini, and Maya Gupta. Launch and iterate: Reducing prediction churn. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pp. 3179–3187, 2016.

Andrei Mircea, Supriyo Chakraborty, Nima Chitsazan, Irina Rish, and Ekaterina Lobacheva. Training dynamics underlying language model scaling laws: Loss deceleration and zero-sum learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28154–28188, 2025.

Taman Narayan, Heinrich Jiang, Sen Zhao, and Sanjiv Kumar. Predicting on the edge: Identifying where a larger model does better. *arXiv preprint arXiv:2202.07652*, 2022.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations (ICLR)*, 2025.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1525–1534, 2016.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems 34*, 2021.

David Picard. torch.manual_seed(3407) is all you need: On the influence of random seeds in deep learning architectures for computer vision. *arXiv preprint arXiv:2109.08203*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019.

Frederik Träuble, Julius von Kügelgen, Matthäus Kleindessner, Francesco Locatello, Bernhard Schölkopf, and Peter V. Gehler. Backward-compatible prediction updates: A probabilistic approach. In *Advances in Neural Information Processing Systems 34*, pp. 116–128, 2021.

Oskar van der Wal, Pietro Lesci, Max Müller-Eberstein, Naomi Saphra, Hailey Schoelkopf, Willem Zuidema, and Stella Biderman. Polypythias: Stability and outliers across fifty language model pre-training runs. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yuqing Xie, Yi-An Lai, Yuanjun Xiong, Yi Zhang, and Stefano Soatto. Regression bugs are in your model! measuring, reducing and analyzing regressions in NLP model updates. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6589–6602, 2021.

Sijie Yan, Yuanjun Xiong, Kaustav Kundu, Shuo Yang, Siqi Deng, Meng Wang, Wei Xia, and Stefano Soatto. Positive-congruent training: Towards regression-free model updates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14299–14308, 2021.

An Yang et al. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024.

An Yang et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

Ruiqi Zhong, Dhruba Ghosh, Dan Klein, and Jacob Steinhardt. Are larger pretrained language models uniformly better? comparing performance at the instance level. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3813–3827, 2021.

A Evaluation setup and robustness

A.1 The corpus

Table 4: Composition of the 120-document evaluation corpus. Position counts are for the Qwen tokenizer used by Qwen2.5 and Qwen3.

genre	source	split	documents	positions
multi-session dialogue	LoCoMo	first 6 conversations	70	35,406
multi-hop QA context	HotpotQA (LongBench)	first 40 contexts	31	14,160
long chat history	LongMemEval	first 40 sessions	19	9,709
total			120	59,275

Table 4 summarizes the evaluation mixture. Every unit from each source is pooled, permuted once with `numpy default_rng(0)`, and the first 120 taken. The three sources are therefore represented in the proportions the pooled draw produced, $70/31/19$, and not equally; Appendix A.6 reports the flows separately by source so that the mixture is not doing any work. Each document is truncated to

its first 512 tokens by the family's own tokenizer, so position counts differ slightly between families (59,275 for Qwen3 and Qwen2.5, 59,390 for Gemma-3); the counts above are for Qwen3. The cross-family analysis instead truncates to a fixed 1,600 characters so both tokenizers see identical text.

Why 512 tokens, and what happens at 1024. Two of the three sources are long-context corpora and could be read much further. A fixed cut is what makes the comparison clean: it equalises the token budget across sources, so the mixture weights above describe the measurement rather than being confounded with document length, and it fits inside every model in the study without a context-length difference entering the comparison. It is also a choice that could set the result, since a model's advantage may grow with context, so we repeat the whole measurement at 1024 tokens on the controlled ladder and the Qwen2.5 ladder, each family using its own tokenizer, and report the flows at both lengths together with the same flows split by position inside the document. Table 5 gives the full 512-versus-1024-token comparison. The cut does not set the result: over the six steps the correctness churn changes by at most 0.18 points between the two lengths, the break rate by at most 0.11, and the directionality by at most 0.026. Nor is the first half of a document different in kind from the second. Averaged over the six steps, C is 10.85, 10.65, 10.95 and 10.71 per cent over the four quarters of a 1024-token document and D is $+0.177$, $+0.182$, $+0.187$ and $+0.184$, so reading further adds positions of the same kind rather than changing what the decomposition says.

Table 5: The flows at the 512-token truncation used throughout and at 1024 tokens, on the controlled ladder and the Qwen2.5 ladder, with the within-document trend below. Reading twice as far does not change what the decomposition says.

	512 tokens				1024 tokens			
step	positions	C	B	D	positions	C	B	D
160M→410M	59244	12.69	4.12	+0.351	95556	12.72	4.11	+0.354
410M→1B	59244	8.97	3.64	+0.189	95556	9.08	3.72	+0.181
1B→1.4B	59244	8.53	3.63	+0.149	95556	8.53	3.65	+0.145
0.5B→1.5B	59275	12.50	4.96	+0.205	94184	12.68	4.87	+0.231
1.5B→3B	59275	10.76	4.78	+0.111	94184	10.74	4.88	+0.091
3B→7B	59275	11.01	5.11	+0.072	94184	10.99	5.00	+0.090
averaged over steps, by position inside the document (at 1024 tokens)								
positions 0–256	182976	10.85	4.42	+0.177				
positions 256–512	173211	10.65	4.33	+0.182				
positions 512–768	129168	10.95	4.40	+0.187				
positions 768–1024	83865	10.71	4.31	+0.184				

A.2 Protocol

Log-probabilities are taken from a single teacher-forced forward pass in float16; no sampling enters the measurement, so the only variation is across models. Greedy predictions are argmax over the full vocabulary.

For the cross-family alignment, each family tokenizes the same character-truncated document with offsets. A token is charged to the word containing the first character it predicts. A word counts as predicted by a model only when every token inside it was predicted, and is compared only when both families covered it. The shuffled null permutes documents on one side, preserving each family's marginal break rate while destroying the correspondence, and is repeated 200 times.

A.3 The central table from the pretrained releases

Table 1 uses the instruction-tuned release of each family, because that is what gets deployed and compared. Post-training is a size-dependent intervention, so on its own that table would not license a claim about pretraining. Here every one of the nine models is replaced by its pretrained release (`Qwen2.5-{0.5,1.5,3,7}B`, `Qwen3-{4,8,14}B-Base`, `gemma-3-{4,12}b-pt`) and run through the same pipeline on both corpora. The flows are the same and slightly smaller: the break rate is 3.30 to 4.82% against 4.01 to 5.22% for the deployed checkpoints, lower in nine of

the twelve cells and by 0.56 points on average, and the directionality of the two agrees at $r = 0.80$. Instruction tuning therefore adds roughly half a point of break rate on top of what pretraining already produces, which is worth knowing and is not what this paper is about.

Table 6 reports the pretrained-release control alongside the instruction-tuned measurements.

Table 6: Table 1 remeasured from each family's *pretrained* release, on the same documents and on held-out WikiText-103, with the instruction-tuned figures repeated for comparison. The flows are a property of the pretrained models and not of post-training, and the second corpus shows the central table is not an artefact of the first.

		pretrained, our corpus						instruction-tuned		pretrained, WT103	
family	step	R	B	net	D	G	$\mathcal{L}$	B	D	B	D
Qwen2.5	0.5B→1.5B	7.66	**4.48**	+3.18	+0.262	0.417	0.177	4.96	+0.205	4.30	+0.328
	1.5B→3B	5.56	**4.15**	+1.40	+0.145	0.262	0.171	4.78	+0.111	4.56	+0.162
	3B→7B	5.21	**4.14**	+1.07	+0.115	0.246	0.166	5.11	+0.072	4.36	+0.231
Qwen3	4B→8B	4.50	**3.62**	+0.88	+0.108	0.212	0.147	4.10	+0.135	4.43	+0.128
	8B→14B	4.81	**3.67**	+1.14	+0.134	0.212	0.147	4.54	+0.060	4.82	+0.125
Gemma-3	4B→12B	5.20	**3.60**	+1.60	+0.182	0.228	0.119	4.93	+0.151	3.30	+0.370

A.4 The central table on a second corpus

Table 1 is measured on one 120-document mixture, which is a fair objection to a claim about next-token prediction in general. Every model in it is rerun here on held-out WikiText-103, through the same pipeline and with each family using its own tokenizer. The break rate moves by at most 0.95 points between the two corpora, so the negative flow is not a property of this corpus. The directionality does move, upwards on WikiText-103 at every step and by up to 0.165: the models are better on encyclopaedic text than on dialogue and long contexts, so the same gross flows cancel less there. The claim the paper rests on is about the flows, which replicate, rather than about the size of the net, which does not have to.

Table 7 gives the full WikiText-103 replication.

Table 7: Table 1 repeated on held-out WikiText-103, with the same models, the same pipeline and each family's own tokenizer. The break rate moves by at most 0.95 points between the two corpora and the directionality by at most 0.165, so Table 1 is not an artefact of the corpus it is measured on.

		our corpus						WikiText-103					
family	step	R	B	net	D	G	$\mathcal{L}$	R	B	net	D	G	$\mathcal{L}$
Qwen3	4B→8B (2.0×)	5.38	**4.10**	+1.28	+0.135	0.418	0.219	6.52	**4.36**	+2.16	+0.199	0.625	0.256
	8B→14B (1.8×)	5.12	**4.54**	+0.58	+0.060	0.320	0.246	6.51	**4.70**	+1.80	+0.161	0.431	0.339
Gemma-3	4B→12B (2.8×)	6.69	**4.93**	+1.75	+0.151	0.573	0.280	8.76	**5.22**	+3.54	+0.253	0.863	0.317
Qwen2.5	0.5B→1.5B (3.1×)	7.53	**4.96**	+2.57	+0.205	0.427	0.207	8.72	**4.01**	+4.71	+0.370	0.577	0.183
	1.5B→3B (2.0×)	5.98	**4.78**	+1.20	+0.111	0.281	0.231	6.11	**4.72**	+1.39	+0.128	0.346	0.241
	3B→7B (2.5×)	5.90	**5.11**	+0.79	+0.072	0.287	0.274	7.01	**4.69**	+2.32	+0.198	0.397	0.251

A.5 Everything on a third corpus

The two corpora used so far are a curated long-context mixture and held-out Wikipedia. Neither is the web text these models were pretrained on, which is a fair thing to ask of a claim about next-token prediction in general, so all nine released models, their pretrained releases and the whole Pythia ladder are repeated on held-out C4. The released break rate is 3.72 to 4.56% for the instruction-tuned checkpoints and 3.06 to 3.75% for the pretrained ones; the data-controlled ladder gives 3.33 to 3.64% across its six steps with the gross likelihood loss again flat, between 0.167 and 0.186 nats, and D falling from $+0.339$ to $+0.103$. Across three corpora the break rate lies between 3.06 and 5.22% and never trends with scale.

Table 8 gives the complete held-out C4 replication.

Table 8: Everything repeated on held-out C4, the web text these models were pretrained on, rather than the curated long-context mixture of Table 1 or the encyclopaedic text of Table 7. The released families are shown for their instruction-tuned checkpoints with the pretrained break rate and directionality beside them, and the data-controlled Pythia ladder follows.

		as measured						pretrained	
suite	step	R	B	net	D	G	$\mathcal{L}$	B	D
Qwen3	4B→8B	5.26	**3.72**	+1.53	+0.171	0.446	0.205	3.12	+0.178
	8B→14B	5.22	**3.96**	+1.26	+0.137	0.345	0.246	3.36	+0.157
Gemma-3	4B→12B	7.00	**4.56**	+2.45	+0.212	0.604	0.272	3.06	+0.272
Qwen2.5	0.5B→1.5B	7.47	**3.84**	+3.64	+0.322	0.499	0.184	3.71	+0.314
	1.5B→3B	5.35	**4.05**	+1.30	+0.138	0.312	0.203	3.73	+0.151
	3B→7B	5.29	**4.28**	+1.01	+0.106	0.310	0.212	3.75	+0.141
Pythia	160M→410M	7.38	**3.64**	+3.74	+0.339	0.556	0.176	—	—
Pythia	410M→1B	5.09	**3.44**	+1.65	+0.193	0.327	0.178	—	—
Pythia	1B→1.4B	4.54	**3.57**	+0.96	+0.118	0.263	0.186	—	—
Pythia	1.4B→2.8B	4.58	**3.36**	+1.22	+0.154	0.280	0.169	—	—
Pythia	2.8B→6.9B	4.32	**3.33**	+0.99	+0.129	0.247	0.167	—	—
Pythia	6.9B→12B	4.24	**3.45**	+0.79	+0.103	0.229	0.170	—	—

A.6 The rates by genre

The corpus mixes three sources in unequal proportion, so the flows are reported separately by source. The same pattern appears in each: both flows are large, the break rate varies by about a point and a half across genres within a step, and the net is positive everywhere. Dialogue is the hardest case for scale: at Qwen3 8B→14B the net there is $+0.12$ points, essentially nothing, because the two flows are 4.69% and 4.58%. Table 9 reports the source-level decomposition.

Table 9: Rescue and break flows by corpus source for the three released-family transitions shown here. The negative flow persists in each source rather than being driven by the pooled mixture.

step	genre	positions	rescued	broken	net top-1	Δ nats
Qwen3 4B→8B	dialogue	35406	5.06%	4.17%	+0.89	+0.178
	multi-hop QA contexts	14160	5.47%	3.45%	+2.01	+0.288
	long documents	9709	5.86%	4.62%	+1.24	+0.147
Qwen3 8B→14B	dialogue	35406	4.69%	4.58%	+0.12	+0.072
	multi-hop QA contexts	14160	5.54%	3.90%	+1.64	+0.090
	long documents	9709	6.25%	5.02%	+1.24	+0.064
Gemma-3 4B→12B	dialogue	35553	5.98%	4.38%	+1.60	+0.260
	multi-hop QA contexts	14128	8.29%	5.46%	+2.83	+0.461
	long documents	9709	6.76%	5.94%	+0.81	+0.170

A.7 What a ceiling effect and a temperature mismatch would look like

A ceiling effect would concentrate the loss where the smaller model is already nearly certain. Between 4.1% and 12.7% of broken positions, and 4.3% to 8.8% of the gross likelihood loss, sit above $p_S(y_{\text{true}}) > 0.9$, so most of the loss is not there. A global temperature rescaling cannot produce the effect either: the greedy prediction is temperature-invariant (Guo et al., 2017), so the top-1 flows are untouched by it, and the larger models here have *higher* entropy than the smaller ones, which is the opposite of what a sharpening mismatch would give. Temperature is one form of calibration mismatch among several and we claim no more than this.

A.8 CROSS-FAMILY OVERLAP AGAINST STRATIFIED NULLS

Do networks built by different groups on different data lose the same content when scaled? Aligning Qwen3 and Gemma-3 on characters and charging every token to the word it starts in, over the $8{,}217$ words both cover, Qwen3 breaks 4.70% and Gemma-3 4.97%, and they break the same word $2.30\times$ more often than independence gives. That needs a null harder than independence, since both families break rare, long, hard words and any two models would. Shuffling within strata that hold difficulty fixed removes most of the excess: the smaller model's confidence decile alone raises the null from 0.99 to 1.52, and adding word length, position and genre, 704 strata in all, raises it to 1.65 ± 0.25, leaving the observed lift at $1.39\times$ the null and 2.6 standard deviations. The shared component is real and modest, and we report it here rather than in the main text because two families is a small basis for a claim about families in general.

Table 10 shows how the overlap changes as the null preserves progressively more difficulty structure.

Table 10: The cross-family break overlap is $2.30\times$ chance, against nulls that preserve progressively more of what makes a word hard. Shuffling within strata of the smaller model's own confidence already accounts for most of the apparent excess, and the fully stratified null leaves $1.39\times$ at 2.6 standard deviations. Part of what scaling gives up is shared beyond difficulty, but it is a small part.

null preserves	strata	null lift	excess	z
per-document break rates only	1	0.99 ± 0.23	$2.32\times$	+5.8
+ smaller model's confidence decile	10	1.52 ± 0.24	$1.51\times$	+3.2
+ word length	59	1.52 ± 0.23	$1.51\times$	+3.4
+ position in document	270	1.57 ± 0.26	$1.46\times$	+2.8
+ genre	704	1.65 ± 0.25	$1.39\times$	+2.6

B CONTROLLED SCALING AND REPLICATION

B.1 THE DATA-CONTROLLED LADDER IN FULL

Table 11 reports all six data-controlled Pythia transitions on both corpora.

Table 11: The data-controlled ladder in full. Pythia trains every size on the same corpus in the same order, so these six steps span a seventy-five-fold range of parameters with the data held fixed. The break rate stays between 3.50 and 4.12% throughout and the gross likelihood loss $\mathcal{L}$ does not fall, while the net contracts from $+4.46$ to $+0.75$ points and the directionality from $+0.351$ to $+0.097$: what diminishes with scale is the gain and the alignment of the churn, not the loss.

	our corpus						WikiText-103					
step	R	B	net	D	G	$\mathcal{L}$	R	B	net	D	G	$\mathcal{L}$
160M→410M	8.58	**4.12**	+4.46	+0.351	0.547	0.151	9.37	**3.85**	+5.52	+0.418	0.722	0.182
410M→1B	5.34	**3.64**	+1.70	+0.189	0.284	0.163	6.17	**3.92**	+2.25	+0.223	0.404	0.196
1B→1.4B	4.90	**3.63**	+1.27	+0.149	0.238	0.171	5.48	**4.03**	+1.46	+0.153	0.321	0.209
1.4B→2.8B	4.73	**3.74**	+0.99	+0.117	0.254	0.147	5.77	**3.92**	+1.85	+0.191	0.337	0.185
2.8B→6.9B	4.40	**3.58**	+0.82	+0.103	0.209	0.149	5.25	**4.08**	+1.18	+0.126	0.293	0.185
6.9B→12B	4.26	**3.50**	+0.75	+0.097	0.190	0.153	5.10	**3.82**	+1.28	+0.143	0.266	0.181

B.2 THE CONTROLLED SIZE LADDER

Table 12 reports conditional transition rates and likelihood flows for the first three controlled size steps.

Table 12: Pythia: every size sees the same data in the same order, so the only thing that changes is parameters. The break rate and the conditional regression rate land in the same range as the released ladders of Table 1, and the pattern is the same — the rescue rate falls steeply while the break rate does not move. Measured on the same documents as Table 1, so the only thing that differs between the two tables is the model suite. Intervals are bootstraps over documents.

step	×	acc. S	acc. L	R	B (95% CI)	net	r_{fix}	r_{reg}	G	$\mathcal{L}$	$P(d{<}0)$
160m→410m	2.5	44.1	48.5	8.58	**4.12** [3.95, 4.28]	+4.46	15.3	**9.3**	0.547	0.151	32%
410m→1b	2.5	48.5	50.2	5.34	**3.64** [3.49, 3.78]	+1.70	10.4	**7.5**	0.284	0.163	42%
1b→1.4b	1.4	50.2	51.5	4.90	**3.63** [3.49, 3.78]	+1.27	9.8	**7.2**	0.238	0.171	44%

B.3 Two trends that are partly arithmetic

The rise in C/Q. The accuracy-relevant share rises from 29.2% at the smallest rung to 39.2% at the largest on our corpus and from 21.2% to 33.8% on WikiText-103. Part of that is forced: as both models improve the cell where both are wrong shrinks, from 61.9% to 48.3% of positions here, and Q falls with it. The part that is not forced is $\rho = W/P(C_S{=}C_L{=}0)$, the share of joint errors on which the two models err *differently*, which falls from 0.507 to 0.362 here and 0.564 to 0.429 on WikiText-103. Holding ρ at its ladder mean and recomputing reproduces C/Q to within 0.9 points at the middle rungs but misses the largest by 4.8, so the shrinking error cell accounts for most of the trend and the decline in ρ for the rest.

The fall in D. The question we put to C/Q has to be put to D as well. R can only draw on the positions the smaller model gets wrong, a pool that shrinks from 55.9% to 46.7% down the ladder, while B draws on the ones it gets right, which grows, so D would fall even if nothing about the models changed. Holding the conditional rates r_{fix} and r_{reg} at their ladder means and recomputing D from the marginals alone reproduces 70% of the observed fall on our corpus and 86% on WikiText-103, with a largest per-step error of 0.064 and 0.058: most of the decline is the shrinking error pool. The residual is that the conditional rates fall too, r_{fix} from 15.3% to 9.1% and r_{reg} from 9.3% to 6.6%, so a larger model fixes a smaller share of what the smaller one gets wrong and spoils a smaller share of what it gets right, which is the same convergence the falling ρ shows.

None of this touches Section 5.3. There the two axes are matched on size and therefore on accuracy, so the pools both flows draw on are identical by construction and no part of the gap in D can come from them.

B.4 The replicated comparison, cell by cell

Table 13 expands the 160M→410M replicated comparison cell by cell.

Table 13: Scaling as a replicated experiment. PolyPythias publishes nine independent runs at 160M and nine at 410M, so pairing them by seed identifier gives nine size transitions rather than one, measured against the same-size pairs those very runs supply. $C = R + B$ is correctness churn and $Q = R + B + W$ is full prediction disagreement, which is two and a half times larger on every axis. $D = (R - B)/C$ is signed for size steps, which are oriented; same-size pairs are exchangeable, so they report $|D|$ with the median over pairs in brackets. The two axes differ by a factor of eleven in direction and by 1.18 in amount: bootstrapping over training runs, ΔC against the 160M baseline is $+1.76$ $[+1.45, +2.13]$ points here and $+1.89$ $[+1.50, +2.26]$ on WikiText-103, and ΔQ is $+2.70$ $[+2.10, +3.29]$ and $+2.57$ $[+2.01, +3.01]$. The gap in direction is $+0.249$ $[+0.120, +0.330]$ and $+0.280$ $[+0.148, +0.368]$.

		our corpus (%)					WikiText-103 (%)				
axis	n	R	B	C	Q	D	R	B	C	Q	D
size step, paired by seed identifier	9	7.20	**4.08**	11.28	28.76	**+0.273**	7.89	**4.16**	12.04	35.67	**+0.303**
run-to-run at 160M	36	4.81	**4.71**	9.52	26.06	**0.024 (0.023)**	5.15	**5.00**	10.15	33.10	**0.023 (0.021)**
run-to-run at 410M	36	5.27	**4.85**	10.12	25.44	**0.151 (0.060)**	5.81	**5.20**	11.01	31.99	**0.160 (0.052)**
more training at 160M	4	4.20	**3.80**	8.01	22.71	**+0.047**	4.46	**4.00**	8.45	28.53	**+0.056**

B.5 Multiplicity, and the arbitrariness of the pairing

Two objections apply to Table 2 as a whole rather than to any row.

Eight intervals at once. The direction gap is reported at four rungs on two corpora. Converting each bootstrap interval to a two-sided p-value through its own standard error and applying a Holm correction at 0.05, the eight tests pass at thresholds from 6.3×10^{-3} to 5.0×10^{-2} with p-values between 7×10^{-110} and 2×10^{-4}, so the claim is not an artefact of testing repeatedly. We do not correct the amount gaps, which are reported as intervals and interpreted as sizes rather than as tests.

Which run is paired with which. Nine runs at one size and nine at the next admit 81 pairings, and the diagonal, pairing seed identifier i with seed identifier i, is one arbitrary choice among them. Averaging over all 81 instead changes nothing that matters: correctness churn moves from 12.98 to 13.10, 12.90 to 12.87, 13.96 to 14.05 and 11.28 to 11.42 per cent over the four rungs, and D from $+0.371$, $+0.311$, $+0.452$, $+0.273$ to $+0.368$, $+0.314$, $+0.449$, $+0.268$. We report the diagonal because it keeps the number of models on each side of the comparison equal to the number of replicates, and note that the full cross-product gives the same answer.

B.6 The ladder plotted

Figure 2 visualizes the amount gap, directional gap, and C/Q trend across the replicated ladder.

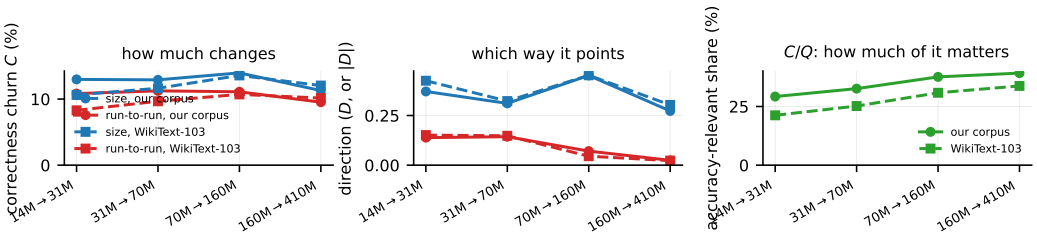

Figure 2: The ladder of Table 2: a size step changes slightly more of the predictions that matter than retraining does at every rung (left), points them far more consistently at every rung (middle), and the share of disagreement that decides a correctness rises with size (right).

B.7 The three axes, pair by pair

Table 14 places size, training-time, and seed changes in the same units.

Table 14: Three ways to change a model, in the same units. Size varies at a fixed checkpoint; checkpoint varies at a fixed size; seed varies neither, only the initialisation and the data order. Seed pairs are unordered, so the sign of their net is arbitrary and only the magnitude means anything. Read off at a net change of one point, the break rate is size 3.56%, checkpoint 3.91%, seed, 160M 4.56%. Changing the seed breaks more than either directed change does.

axis	pair	acc. A	acc. B	net	rescued	broken	changed
size	160m→410m	44.1	48.5	+4.46	8.58	**4.12**	12.69
	410m→1b	48.5	50.2	+1.70	5.34	**3.64**	8.97
	1b→1.4b	50.2	51.5	+1.27	4.90	**3.63**	8.53
checkpoint	step 13000→step 26000	42.5	43.5	+1.00	4.83	**3.83**	8.66
	step 26000→step 52000	43.5	44.0	+0.50	4.41	**3.92**	8.33
	step 52000→step 104000	44.0	44.1	+0.15	4.12	**3.97**	8.09
	step 104000→160m	44.1	44.1	-0.04	3.45	**3.49**	6.94
seed, 160M	mean of 36 pairs	–	–	+0.11	4.81	**4.71**	9.52
seed, 410M	mean of 36 pairs	–	–	+0.41	5.27	**4.85**	10.12

B.8 TRAJECTORIES OVER THE SIZE LADDER

Table 15 quantifies how often broken predictions recover at the next rung.

Table 15: A prediction broken at one size step, seen again one step further up the Qwen2.5 ladder. Roughly half return, against a base rate of one in eight for an arbitrary position the middle model gets wrong, so most breaks are churn rather than permanent loss. A residue does not return: of the positions 0.5B gets right, 17.9% are wrong at some larger size and 11.2% are still wrong at 7B, and 14.0% of all trajectories are non-monotone.

broken at	seen at	n	recovered	95% CI	base rate	lift
0.5B→1.5B	3B	2910	43.4%	[41.6, 45.3]	12.4%	3.49×
1.5B→3B	7B	2811	48.3%	[46.6, 50.1]	12.8%	3.77×

B.9 A FROM-SCRATCH TRAINING RUN

If a larger model improves by breaking as well as fixing, the natural question is whether that is something scale does or something improvement does. Pretrained models cannot answer it, since they arrive finished, so we trained one: a 51.9M parameter decoder on 120M tokens of WikiText-103, checkpointed eight times, and ran the same decomposition between consecutive checkpoints on held-out text.

Table 16 follows the rescue and break flows over the full from-scratch training run.

Table 16: One 51.9M model, eight checkpoints, the same decomposition between consecutive ones on held-out text. Learning breaks what it had at every point; at convergence the two flows equalise at 0.20% and the net is zero.

tokens seen	NLL	top-1	rescued	broken	net
5%	5.6782	20.52%	–	–	–
10%	5.2914	23.51%	5.64%	2.64%	+3.00
20%	4.8525	26.98%	6.01%	2.55%	+3.46
35%	4.5857	29.41%	4.88%	2.44%	+2.44
50%	4.4669	30.88%	3.47%	2.00%	+1.46
70%	4.4064	31.80%	2.31%	1.38%	+0.93
85%	4.3988	31.82%	0.76%	0.74%	+0.02
100%	4.3985	31.82%	0.20%	0.20%	-0.01

Learning trades. At every point in the run the model breaks predictions it previously had, and the improvement that gets reported is again the difference. What differs from scale is the trajectory: the churn *decays*. Early in training the model rescues 5.6% and breaks 2.6%; by the end it rescues 0.20% and breaks 0.20%, and the net is zero. Convergence here is not the model running out of things to fix. It is the two flows becoming equal.

Scale trades harder than training. The two axes can be put on the same footing by matching them on what they deliver. Between 20% and 35% of training this model gains +2.44 points of top-1 accuracy and breaks 2.44% of positions. Going from Qwen2.5-0.5B to Qwen2.5-1.5B gains a comparable +2.70 points and breaks 4.91%. For the same net improvement, changing the size costs about twice the damage that continuing to train does.

We state that comparison carefully, and it is why this run is reported in an appendix rather than as a result. A 51.9M model on WikiText-103 and a pair of released multi-billion parameter models on web text differ in far more than the axis being varied, and the numbers are matched on outcome rather than on anything else. What the run supports is the qualitative claim, that both axes trade and the churn decays along one and not the other, rather than an exchange rate. The controlled version of the comparison, at a single size within one suite, is Section 5.

C BENCHMARK-LEVEL EVALUATIONS

C.1 TASK-LEVEL FLIPS IN FULL

Table 17 reports the benchmark-level decomposition for all evaluated tasks and steps.

Table 17: Task-level flows on the whole of ARC-Challenge and on 2,000 examples each of HellaSwag and LAMBADA, rather than the 400-example slice used in earlier drafts. Benchmark examples are independent, so intervals are a bootstrap over examples. The break rate is resolved well away from zero at every step.

suite and task	step	n	acc S	acc L	rescued (%)	broken (%)	net
Qwen2.5, ARC-Challenge	0.5B→1.5B	1172	33.4	46.7	19.37 [17.06,21.67]	**6.14** [4.78,7.51]	+13.23 [+10.41,+16.13]
	1.5B→3B	1172	46.7	46.2	9.64 [7.94,11.43]	**10.15** [8.36,11.95]	-0.51 [-3.07,+2.13]
	3B→7B	1172	46.2	53.4	14.59 [12.63,16.64]	**7.34** [5.89,8.87]	+7.25 [+4.69,+9.90]
Qwen2.5, HellaSwag	0.5B→1.5B	2000	39.7	47.2	12.40 [10.95,13.90]	**4.85** [3.90,5.80]	+7.55 [+5.75,+9.35]
	1.5B→3B	2000	47.2	51.7	9.95 [8.65,11.25]	**5.45** [4.50,6.40]	+4.50 [+2.75,+6.20]
	3B→7B	2000	51.7	57.2	10.65 [9.30,12.00]	**5.20** [4.20,6.25]	+5.45 [+3.65,+7.15]
Qwen2.5, LAMBADA	0.5B→1.5B	2000	50.5	60.1	16.75 [15.10,18.50]	**7.15** [6.05,8.35]	+9.60 [+7.45,+11.70]
	1.5B→3B	2000	60.1	66.5	12.90 [11.40,14.45]	**6.50** [5.40,7.65]	+6.40 [+4.45,+8.35]
	3B→7B	2000	66.5	71.0	11.50 [10.15,12.90]	**6.95** [5.90,8.05]	+4.55 [+2.80,+6.45]
Pythia, ARC-Challenge	160M→410M	1172	23.0	24.4	7.76 [6.23,9.30]	**6.40** [5.03,7.85]	+1.37 [-0.77,+3.50]
	410M→1B	1172	24.4	27.0	7.59 [6.06,9.13]	**4.95** [3.67,6.23]	+2.65 [+0.60,+4.69]
	1B→1.4B	1172	27.0	28.7	7.94 [6.48,9.56]	**6.31** [4.95,7.76]	+1.62 [-0.60,+3.75]
Pythia, HellaSwag	160M→410M	2000	29.9	34.5	9.30 [8.05,10.60]	**4.70** [3.80,5.60]	+4.60 [+3.00,+6.25]
	410M→1B	2000	34.5	37.5	7.15 [6.00,8.35]	**4.25** [3.40,5.15]	+2.90 [+1.45,+4.40]
	1B→1.4B	2000	37.5	39.9	6.75 [5.65,7.85]	**4.35** [3.45,5.25]	+2.40 [+0.95,+3.85]
Pythia, LAMBADA	160M→410M	2000	34.2	51.6	21.30 [19.50,23.10]	**3.95** [3.10,4.80]	+17.35 [+15.25,+19.40]
	410M→1B	2000	51.6	56.8	11.95 [10.50,13.35]	**6.80** [5.70,7.90]	+5.15 [+3.25,+7.00]
	1B→1.4B	2000	56.8	62.6	11.30 [9.95,12.70]	**5.40** [4.45,6.40]	+5.90 [+4.10,+7.65]

D ROUTING CERTIFICATE AND CONFIDENCE DIAGNOSTICS

D.1 PROOF OF THE ROUTING BOUND

We restate Eq. (4) and prove both steps. Let s be a statistic of the smaller model's output at a position, $a = P(R)$ and $b = P(B)$ the rescue and break probabilities, and f_R, f_B the conditional densities of s given a rescue and given a break. Consider any escalation rule measurable with respect to s, that is, any set $E \in \sigma(s)$.

Step 1: the optimal measurable set. Escalating E replaces the smaller model's prediction with the larger model's exactly on E. Accuracy changes only at rescued and broken positions, gaining at the first and losing at the second, so

$$\Delta(E) = \int_E \big(a f_R(s) - b f_B(s)\big)\, ds.$$

The integrand is a fixed signed function of s, so Δ is maximised by including exactly the region where it is positive: $E^\star = \{s : a f_R(s) > b f_B(s)\}$ and $G^\star = \int \max(a f_R - b f_B, 0)\, ds$. Threshold rules $\{s \geq t\}$ and any rule of the form $\{g(s) \geq t\}$ are members of $\sigma(s)$, so $G^\star$ bounds them all; so does any classifier whose inputs are functions of s, since it generates no finer σ-algebra.

Step 2: the bound. For real u, $\max(u, 0) = \frac{1}{2}(u + |u|)$. Applying this pointwise and integrating, and using $\int (a f_R - b f_B) = a - b$,

$$G^\star = \tfrac{a-b}{2} + \tfrac{1}{2} \int |a f_R - b f_B|\, ds.$$

Write $a f_R - b f_B = a(f_R - f_B) + (a - b) f_B$. By the triangle inequality, $\int |a f_R - b f_B| \leq a \int |f_R - f_B| + |a - b| = 2a\, \mathrm{TV}(f_R, f_B) + |a - b|$, since $\mathrm{TV}(f_R, f_B) = \frac{1}{2} \int |f_R - f_B|$. For $a \geq b$ this gives

$$G^\star \;\leq\; \tfrac{a-b}{2} + a\, \mathrm{TV}(f_R, f_B) + \tfrac{a-b}{2} \;=\; (a - b) + a \cdot \mathrm{TV}(f_R, f_B),$$

which is Eq. (4). The first term is the gain from escalating every position, so $a \cdot \mathrm{TV}$ bounds everything selective routing on s can add. The bound is tight when f_B is concentrated where f_R is not, and Table 18 reports both TV and the exact $G^\star$, so the looseness is visible rather than assumed.

A precision trap worth naming. Gemma-3 loaded in float16 predicts the padding token at every position and scores zero accuracy: its activations overflow the format, and nothing in the stack reports an error. We run it in bfloat16, which reproduces the released model's behaviour, and mention it because a silent zero is the kind of failure a per-token study can carry a long way before anyone notices.

What the estimate is, and why it carries a null. TV above is a population quantity. A plug-in estimate from two histograms is biased upwards: two samples from the *same* distribution give a positive value whose size depends on the two sample sizes and on the binning, not on any real difference. We estimate it by binning s into 40 equal-width bins between its 0.1 and 99.9 percentiles and summing $\frac{1}{2}\sum_k |\hat{f}_R(k) - \hat{f}_B(k)| w_k$. This plug-in estimator is biased upwards, by an amount that depends on the two sample sizes and the binning and not on any real difference between the distributions, so we report it against the null obtained by permuting the rescue and break labels 200 times among the positions carrying either, which is the value the same estimator returns when the two distributions are identical.

D.2 The routing certificate in full

Table 18 compares the separability of entropy, peer disagreement, and an answer-aware oracle score.

Table 18: Total variation between the distributions of a statistic on rescued and on broken positions, which is what Eq. (4) converts into a ceiling on routing. A plug-in estimate from two histograms is biased upwards, so each is shown against the value the same estimator returns when the rescue and break labels are permuted at these sample sizes: that null is what a statistic carrying no information scores. Entropy barely clears its own null and on one step does not clear it at all. Peer disagreement clears it by an order of magnitude, and the answer-aware score, which needs the label and is not deployable, by two.

	entropy			peer disagreement α			answer-aware $-\log p_S(y)$		
step	$\widehat{\mathrm{TV}}$	null	excess	$\widehat{\mathrm{TV}}$	null	excess	$\widehat{\mathrm{TV}}$	null	excess
Qwen2.5 1.5B→3B	0.070	0.052	**+0.018**	0.133	0.010	**+0.123**	0.490	0.034	**+0.456**
Qwen2.5 3B→7B	0.049	0.050	≈ 0	0.145	0.010	**+0.134**	0.511	0.031	**+0.480**
Qwen3 8B→14B	0.052	0.043	**+0.009**	0.102	0.010	**+0.092**	0.578	0.033	**+0.546**

Table 19 evaluates the routing certificate exactly and compares it with held-out thresholds.

Table 19: The certificate evaluated exactly. $G^\star$ is the optimum $\int \max(af_R - bf_B, 0)$ over all sets measurable in s; the bound is $(R-B) + R \cdot \widehat{\mathrm{TV}}(f_R, f_B)$; held-out is the best threshold fitted on half the documents and scored on the other half. Entropy separates rescued from broken positions almost not at all, so it buys a few hundredths of a point over escalating everything, and the fitted threshold recovers essentially all of the little there is. The answer-aware score, which needs the label, separates them and buys points.

		entropy (available at inference)				$-\log p_S(y)$ (answer-aware)		
step	$R-B$	$\widehat{\mathrm{TV}}$	$G^\star$	bound	held-out	$\widehat{\mathrm{TV}}$	$G^\star$	held-out
Qwen3 4B→8B	+1.21	0.083	1.25	1.65	+1.14	0.633	3.70	+3.67
Qwen3 8B→14B	+0.66	0.053	0.78	0.94	+0.74	0.600	3.23	+3.26
Gemma-3 4B→12B	+1.76	0.072	1.79	2.25	+1.77	0.723	5.07	+5.11

D.3 Both flows by entropy bin

Figure 3 shows that rescue and break mass move together across entropy.

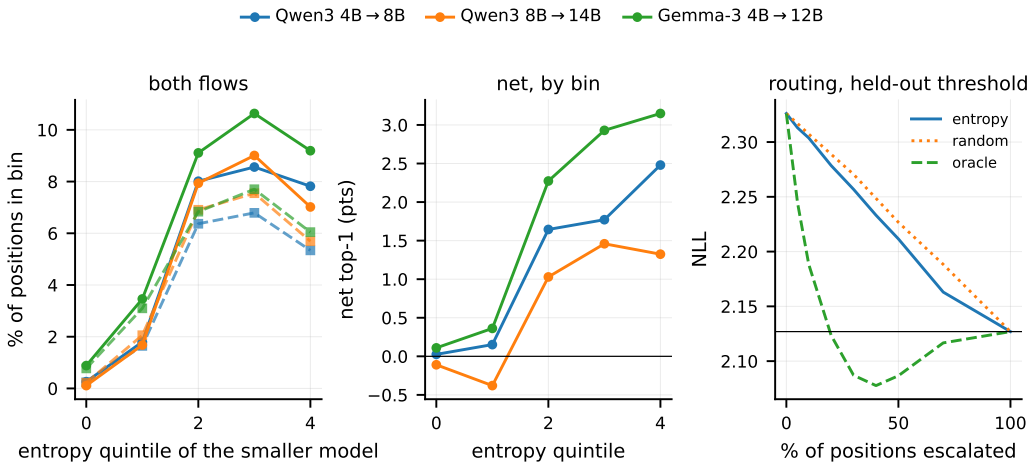

Figure 3: Both flows through the smaller model's own entropy, and the routing curve with the threshold fitted on held-out documents. The flows rise and fall together, which is why no threshold separates them; the deployable router tracks the random one.

Table 20 gives the corresponding entropy-bin decomposition.

Table 20: Positions binned by the smaller model's own entropy, which it computes without the answer. In the two most confident fifths the larger model is at best neutral and for Qwen3 8B→14B a net loss. The share of all rescues falling in the confident 40% is 6.9–13.1%, so a router that escalates there spends most of its budget on positions with nothing to gain.

entropy quintile	mean H	acc. of S	rescued	broken	net
Qwen3 4B→8B					
0	0.02	96.1%	0.26%	0.24%	+0.03
1	0.33	71.7%	1.80%	1.64%	+0.15
2	0.88	48.1%	8.01%	6.37%	+1.64
3	1.49	31.1%	8.56%	6.79%	+1.77
4	2.74	17.6%	7.82%	5.34%	+2.48
Qwen3 8B→14B					
0	0.02	97.5%	0.11%	0.22%	-0.11
1	0.36	74.1%	1.68%	2.06%	-0.38
2	0.96	48.6%	7.94%	6.91%	+1.03
3	1.62	32.5%	9.01%	7.55%	+1.46
4	2.95	18.0%	7.02%	5.69%	+1.32
Gemma-3 4B→12B					
0	0.01	94.6%	0.88%	0.77%	+0.11
1	0.27	70.8%	3.46%	3.10%	+0.36
2	0.82	49.7%	9.11%	6.84%	+2.27
3	1.48	33.8%	10.63%	7.70%	+2.93
4	2.64	18.5%	9.19%	6.04%	+3.15

D.4 Other confidence statistics, and a learned one

The controlled suite records $p_{\max}$ and the top1–top2 margin alongside entropy. All three give $\widehat{\mathrm{TV}}$ between 0.05 and 0.11 against 0.43 to 0.50 for the answer-aware score, and $G^\star$ exceeds $R - B$ by at most 0.07 points against 1.03 to 1.26. The margin, which is what one would reach for if breaks were near-ties, is the worst of the three at the first step. Fitting does not help: a gradient-boosted classifier trained on half the documents to predict rescue-versus-break from entropy, position, genre and the log-probability of the model's own top-1, applied at its best threshold on the other half, gains $+1.16$, $+0.70$ and $+1.24$ points against the $+1.21$, $+0.66$ and $+1.76$ of escalating every position, and loses to always-large in NLL in all three.

D.5 Escalation budgets

The table the deployable-routing claim in Section 6 summarises, stated in nats.

Table 21 gives the cross-fitted NLL at several escalation budgets.

Table 21: Escalation NLL with the threshold fitted on a disjoint half of the documents and reported on the other half. Routing on the smaller model's own entropy never beats calling the larger model everywhere at any budget below 100%, and is not separated from a random router at matched budget. Only ordering by $-\log p_S(y_{\text{true}})$, which requires the answer, does better; it is an upper bound, not a policy.

			entropy router, % escalated				random	oracle
step	small only	large only	10%	20%	30%	50%	best	best
Qwen3 4B→8B	2.3261	2.1270	2.3037	2.2785	2.2568	2.2116	2.1883	2.0777
Qwen3 8B→14B	2.1295	2.0518	2.1267	2.1198	2.1154	2.0929	2.0765	1.9959
Gemma-3 4B→12B	2.4290	2.1354	2.4027	2.3753	2.3432	2.2780	2.2310	2.0480

E Consensus routing and operating limits

E.1 The cascade at benchmark granularity

Table 22 reports the benchmark-level cascade and its cross-fitted operating point.

Table 22: The same cascade scored the way a benchmark is scored, with its operating point chosen on a disjoint half of the examples rather than carried over from the token-level setting. It escalates 89% of examples on average, which is to say it mostly declines to route, and lands within 0.23 points of always-large. Pinning it to the token-level policy of escalating the whole disagreement set would cost 2.46 points, and the ceiling column shows there was almost nothing to win either way. S/L is how often the deployed model is right where the two disagree, relative to the larger model.

						difference from always-large (points)			
suite	task	step	peer	S/L	always	esc.	fitted	pinned	ceiling
Qwen2.5	ARC-Challenge	1.5B→3B	0.5B	1.05	-0.51	50%	**-0.09**	-1.96	+2.13
	HellaSwag	1.5B→3B	0.5B	0.55	+4.50	100%	**+0.00**	-3.70	+0.00
	LAMBADA	1.5B→3B	0.5B	0.50	+6.40	100%	**+0.00**	-1.30	+0.00
	ARC-Challenge	3B→7B	0.5B	0.50	+7.25	100%	**+0.00**	-6.40	-0.17
	HellaSwag	3B→7B	0.5B	0.49	+5.45	100%	**+0.00**	-4.25	-0.25
	LAMBADA	3B→7B	0.5B	0.60	+4.55	71%	**-0.15**	+0.10	-0.15
Pythia	ARC-Challenge	410M→1B	160M	0.65	+2.65	100%	**+0.00**	-1.62	-0.26
	HellaSwag	410M→1B	160M	0.59	+2.90	100%	**-0.00**	-2.85	+0.70
	LAMBADA	410M→1B	160M	0.57	+5.15	100%	**+0.00**	-1.85	+0.00
	ARC-Challenge	1B→1.4B	410M	0.80	+1.62	50%	**-2.56**	-1.54	-1.19
	HellaSwag	1B→1.4B	410M	0.64	+2.40	100%	**+0.00**	-2.20	-0.40
	LAMBADA	1B→1.4B	410M	0.48	+5.90	100%	**+0.00**	-2.00	+0.00
mean over the twelve steps				0.62	+4.02	89%	**-0.23**	-2.46	+0.03

E.2 Arbitrating after the larger model has answered

A rule that decides before the larger model is called can save compute; one that decides after cannot, but is free to use more. The *arbiter* calls the larger model on every position and then uses every cheaper rung of the ladder together with that model's own entropy to decide whether to keep its answer, with the cell decisions fitted on a disjoint half of the documents. Cross-fitted it gains $+0.26$ $[+0.09, +0.43]$, $+0.63$ $[+0.43, +0.82]$ and $+0.30$ $[+0.15, +0.45]$ points over always-large at 3.3 to $4.1\times$ the deployed model's cost, against $+0.23$, $+0.37$ and $+0.18$ for the cheap router. The policy class contains always-large because escalating every position is one of the available choices; cross-fitted selection can nevertheless deviate slightly on held-out data because the selected policy is estimated from a disjoint split. The cross-family peer of Appendix E.3 reaches the same range of gains at 2.4 to $3.0\times$, which makes the arbiter the more expensive way to buy them.

E.3 Peers from another family

Table 23 compares same-family and cross-family peers directly.

Table 23: Where the peer comes from. Qwen2.5 and Qwen3 tokenize identically, so a model from one can stand as a peer for a deployed model from the other on exactly the same positions, and the two families were trained separately and a generation apart. A peer from the other family separates the flows better than a cheaper rung of the deployed model's own ladder on every configuration: total variation clears its permutation null by 0.166 to 0.208 against 0.092 to 0.134, and the cascade beats always-large by $+0.60$ to $+0.78$ points against $+0.18$ to $+0.39$. Such a peer is not cheap, so this is a statement about the signal rather than a deployment proposal.

step	peer	origin	esc.	cost	$\widehat{\mathrm{TV}}$	excess	gain	always	vs always
3B→7B	0.5B	same family	31%	1.93	0.135	+0.125	+1.23	+0.87	**+0.37**
	1.5B	same family	24%	2.10	0.145	+0.134	+1.26	+0.87	**+0.39**
	Qwen3-4B	cross family	28%	2.98	0.176	+0.166	+1.47	+0.87	**+0.60**
Qwen3-8B→Qwen3-14B	Qwen3-4B	same family	21%	1.87	0.102	+0.092	+0.85	+0.66	**+0.18**
	7B	cross family	27%	2.41	0.219	+0.208	+1.45	+0.66	**+0.78**
	3B	cross family	27%	1.86	0.214	+0.204	+1.41	+0.66	**+0.75**

E.4 When the cascade pays for itself

Which cheaper rung to use is a design choice, and we make it on the same half of the documents that sets the threshold rather than on the half the method is scored on. Only Qwen2.5 3B→7B offers a choice: the fitting half prefers 1.5B ($+1.47$ points) to 0.5B ($+1.43$), within noise of each other, which is worth knowing since the cheaper rung costs a third as much to serve.

Serving the deployed model with peers and escalating a fraction ϕ costs $s + \sum_j p_j + \phi b$ forward-pass units against b for calling the larger model everywhere, so the cascade is cheaper exactly when $b/s > (1 + \sum_j p_j/s)/(1 - \phi)$. Two of the three released configurations satisfy it: 1.93 units against 2.47 for Qwen2.5-3B→7B and 1.89 against 2.01 for 1.5B→3B. The third, Qwen3-8B→14B, costs 1.87 against 1.80, 4% dearer, and gains $+0.18$ points more. The condition also says when the method cannot pay at all: on the controlled suite the step is only $2.5\times$ and the only peers available are full-size independent runs rather than cheaper rungs, so $\sum_j p_j/s \geq 1$ and no escalation fraction satisfies the inequality. The signal is general; the economics need a ladder with a cheap rung on it.

E.5 The ceiling for every rule built from these models

Table 24 reports the resulting ceiling at token and benchmark granularity. How the ceiling is computed: a router commits before seeing the larger model's answer; a rule that waits may *arbitrate* instead, keeping the deployed model's answer where the peers back it, which is measurable with respect to a finer algebra and so escapes Eq. (6). The best rule of that kind partitions the disagreement set by everything it can observe and keeps the deployed model exactly on the cells where it is right more often, so the ceiling follows from listing cells rather than searching over rules. One column below gives the reason both regimes behave as they do: where the two models disagree the smaller is right 0.81 to 0.87 times as often as the larger at token level but 0.62 times on the benchmarks.

Table 24: What *any* rule built from the deployed model, the larger model and the peers can gain over calling the larger model everywhere. The best such rule partitions the disagreement set by whatever it can observe and keeps the deployed model exactly on the cells where that model is right more often, so the ceiling follows from listing cells; cross-fitted decides the cells on half the data and scores on the other. The column that explains everything is S/L, the ratio of how often each model is right where they disagree: near parity at token level, where a ceiling exists and our rule attains it, and 0.62 on average across the benchmarks, where the ceiling itself is $+0.03$. *always* is what calling the larger model everywhere gains over the deployed model, and *attained* is our fitted rule, both in points.

		where the two models disagree (%)				gain over always-large (pts)			
setting	step	of all	S right	L right	S/L	always	ceiling	cross-fitted	attained
token level	1.5B→3B	24.3	19.5	24.1	0.81	+1.12	+0.23	+0.23	**+0.23**
token level	3B→7B	24.7	20.4	23.9	0.85	+0.87	+0.53	+0.46	**+0.53**
token level	8B→14B	21.0	21.4	24.6	0.87	+0.66	+0.18	+0.18	**+0.18**
Qwen ARC-Challenge	1.5B→3B	29.4	34.6	32.8	1.05	-0.51	+2.47	+2.13	**-0.09**
Qwen HellaSwag	1.5B→3B	20.9	26.0	47.5	0.55	+4.50	+0.00	+0.00	**+0.00**
Qwen LAMBADA	1.5B→3B	29.0	22.4	44.5	0.50	+6.40	+0.00	+0.00	**+0.00**
Qwen ARC-Challenge	3B→7B	32.3	22.7	45.1	0.50	+7.25	+0.77	-0.17	**+0.00**
Qwen HellaSwag	3B→7B	21.9	23.8	48.7	0.49	+5.45	+0.30	-0.25	**+0.00**
Qwen LAMBADA	3B→7B	26.9	25.8	42.8	0.60	+4.55	+0.45	-0.15	**-0.15**
Pyth ARC-Challenge	410M→1B	22.7	21.8	33.5	0.65	+2.65	+0.00	-0.26	**+0.00**
Pyth HellaSwag	410M→1B	19.4	21.9	36.8	0.59	+2.90	+0.70	+0.70	**-0.00**
Pyth LAMBADA	410M→1B	31.5	21.6	37.9	0.57	+5.15	+0.00	+0.00	**+0.00**
Pyth ARC-Challenge	1B→1.4B	25.0	25.3	31.7	0.80	+1.62	+0.34	-1.19	**-2.56**
Pyth HellaSwag	1B→1.4B	17.8	24.4	37.9	0.64	+2.40	+0.50	-0.40	**+0.00**
Pyth LAMBADA	1B→1.4B	27.3	19.8	41.4	0.48	+5.90	+0.25	+0.00	**+0.00**
mean over the twelve benchmark steps		25.3	24.2	40.1	0.62	+4.02	+0.48	+0.03	**-0.23**

Peers weaker than the deployed model carry evidence that a prediction is unsettled, not evidence that it is right, and that cannot overturn a posterior that already favours the larger model two to one. Beating always-large on the benchmark steps would need a peer of the larger model's own class, which costs at least what it saves.

F LIMITATIONS

F.1 LIMITATIONS, AND WHAT WE COULD NOT RULE OUT

Scale. The data-controlled measurement reaches 12B (Appendix B.1) and the released families 14B, but the *replicated* comparison, which is what separates scale from retraining, rests on models between 14M and 410M, because 410M is the largest size at which many independent pretraining runs have been published. Whether the ratio between direction and amount holds at the scales practitioners deploy is not a question released artifacts can currently answer, and we do not claim it does.

Which checkpoints Table 1 uses. The three released families are their instruction-tuned releases, because those are the models that get deployed and compared. A size step there moves parameters, pretraining data and a post-training stage that differs by size, so we draw no causal conclusion from that table alone; Appendix A.3 repeats it from the pretrained releases and finds the same flows, half a point smaller.

Corpus. Table 1 is measured on a single 120-document mixture of dialogue, multi-hop question-answering contexts and long chat histories. It is replicated on held-out WikiText-103 (Appendix A.4) and on held-out C4 (Appendix A.5), reported separately by source (Appendix A.6) and unchanged at 1024 tokens (Appendix A.1). Three corpora are not many, and all three are English.

What the routing result is and is not. The cascade is evaluated on greedy next-token accuracy, which is the substrate of generation and not a deployment metric. Where it is decisively better is exactly where the step is nearly a wash, and Eq. (6) makes that a prediction rather than an excuse: on multiple-choice benchmarks, whose steps are three times more directional, no rule buildable from these models beats calling the larger model everywhere, and we report that rather than omitting it. Whether the token-level gain propagates to generation quality under sampling is untested here.

Design choices we did not vary. The primary deployment experiments use cheaper rungs of the same family as peers. We additionally test cross-family peers as a signal-quality control (Appendix E.3), but these peers are not always cheaper and therefore do not constitute the main deployment setting. The benchmarks are scored in the conventional way, length-normalised log-probability for the multiple-choice sets and exact match on greedy generation for LAMBADA, and HellaSwag and LAMBADA are subsampled to 2,000 examples.

The confound we cannot remove inside a released family. The sizes within a released family are checkpoints that differ in training tokens and data mix as well as in parameters, so Table 1 is a comparison of the models practitioners deploy rather than a controlled sweep; that the effect holds across three released families from two organizations is why we report it as more than a property of one recipe, but the controlled statement in Section 5 is the one the claims rest on.