DATA–TASK ALIGNMENT AS A THIRD SCALING AXIS FOR LANGUAGE-MODEL FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Scaling laws for language models usually predict loss from model size and training tokens while treating the data distribution as fixed. This assumption is especially limiting in fine-tuning, where examples at the same token budget can differ substantially in their relevance to the target task. Existing laws cannot quantify that difference. We introduce *data–task alignment* α, a measurable property of a training set relative to a held-out target, and study it jointly with parameter count N and fine-tuning tokens T. Our primary experiment contains 51 controlled Qwen3 cells across four model sizes, six alignment conditions, and a $21\times$ token range. Within this grid, the first-order law $L(\alpha, N, T) = a + bN^{-\beta} + cT^{-\gamma} - d\alpha$ predicts held-out loss with $R^2 = 0.993$ and RMSE 0.015. Omitting alignment changes both fitted compute exponents, so heterogeneous data composition can be misattributed to scaling with N and T. We then map the law's operating regime through a crossed-budget study that finds mild dependence on the token-to-parameter ratio. Selection beyond Q90 shows diminishing returns and eventual reversal as the eligible pool narrows. The alignment effect replicates on an independent math target, a code domain, Llama-3.2, and a frozen gradient selector, although the preferred functional form changes. Alignment therefore provides a useful third scaling coordinate whose interpretation remains tied to the target and measurement. These results connect task-specific data selection to finite-budget scaling by showing when better-targeted data can substitute for compute. They also identify how this return changes across regimes and eventually reverses when selection becomes too narrow.

1 INTRODUCTION

Empirical scaling laws make expensive training runs easier to plan. They relate loss to parameter count and training tokens, then extrapolate across compute (Hestness et al., 2017; Kaplan et al., 2020; Hoffmann et al., 2022). In these laws, the data distribution is usually a fixed background variable. In fine-tuning, it is often the main decision. A practitioner chooses which examples to train on, which target distribution to prioritize, and whether a smaller but more relevant subset is preferable to a larger generic one.

This choice becomes more consequential as fine-tuning pools grow. Additional tokens are valuable only to the extent that their training signal transfers to the target distribution. Two runs can therefore use the same architecture, optimizer, and token budget while producing different returns because their examples cover the target differently. Treating composition as fixed hides this degree of freedom precisely when practitioners need to allocate limited fine-tuning compute.

Existing work shows that data selection can improve language models at fixed budget (Xie et al., 2023a;b; Xia et al., 2024; Liu et al., 2024b). That result does not yet quantify how much an alignment increase is worth at a different model size or token budget. Conversely, a law in N and T cannot distinguish two equal-sized fine-tuning sets whose relevance to the target differs. We ask:

Can target alignment become a measured scaling coordinate alongside model size and training tokens, and over what regime does that coordinate predict fine-tuning loss?

We study full supervised fine-tuning with a policy-independent alignment score. Each candidate example receives a scalar score against a held-out target using embedding and compression similarity.

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

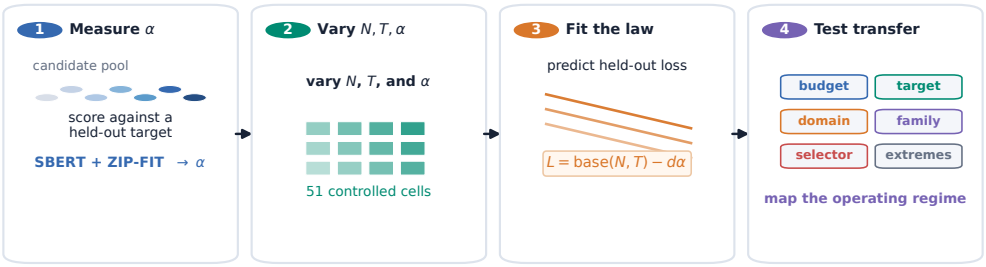

Figure 1: Data–task alignment becomes a measured scaling coordinate with an empirically mapped operating regime. We score examples relative to a held-out target, vary alignment jointly with model size and token budget, fit a compact in-range law, and map its boundaries across budgets, targets, domains, selectors, and extreme selection.

We then construct controlled subsets, train across model sizes and token budgets, and fit loss as a function of the realized subset-level alignment. Figure 1 summarizes the study.

The primary 51-cell math grid admits a simple first-order description:

$$L(\alpha, N, T) = a + \frac{b}{N^{\beta}} + \frac{c}{T^{\gamma}} - d\alpha. \tag{1}$$

The fit reaches $R^2 = 0.993$ and RMSE 0.015. This law is deliberately compact. The new experiments locate its validity. Its additive alignment term is a strong approximation inside the original range. A crossed-budget study finds a mild interaction with T/N, and extreme selection reverses the trend after the candidate pool becomes narrow. A more flexible saturating curve also improves in-sample fit, although its curvature is not stable under cell bootstrap. Together, these observations establish alignment as a third scaling axis with an empirically measurable operating regime.

Three distinctions organize the paper. Alignment is a coordinate because each subset has a measured α, just as every run has measured N and T. Equation (1) is a first-order approximation because it uses one local alignment slope d across the original grid. Its operating regime is the range of model sizes, token budgets, and selected-data support over which that approximation remains predictive. Here, support means the breadth and variety of eligible examples, and support collapse occurs when extreme filtering narrows that pool enough to reverse the expected gain. Both the coordinate and its return remain tied to the target, scoring rule, and training regime.

Our contributions are:

1. a controlled 51-cell study that jointly varies target alignment, parameter count, and fine-tuning tokens.
2. a compact first-order law that predicts loss across the measured grid, quantifies alignment in compute-equivalent terms, and shows how omitting it changes fitted compute scaling.
3. crossed-budget and extreme-quantile experiments that identify weak resource dependence, diminishing returns, and eventual reversal.
4. replication across an independent target, code, Llama-3.2, and a frozen gradient-based selector, separating the value of a measured alignment coordinate from the universality of any one score or functional form.

2 Related Work

Scaling model size, tokens, and data composition. Classical language-model scaling laws relate loss to model size, data, and compute (Kaplan et al., 2020; Hoffmann et al., 2022). Fine-tuning studies likewise find systematic interactions among model scale, data size, and tuning method (Zhang et al., 2024). Broken and rectified laws model changes of regime that a single power law cannot capture (Caballero et al., 2022; Lin et al., 2024). Most relevant to our setting, Hashimoto (2021) predicts performance under mixtures of named data sources and models composition through the intercept of

a learning curve. Data Mixing Laws instead fit predictive response surfaces over mixture proportions (Ye et al., 2025). We move from named-source mixtures to example-level target alignment, jointly vary N and T, and measure where a compact alignment term ceases to transfer.

Task-specific data selection. Data curation and pruning can improve performance at fixed token count (Sorscher et al., 2022; Penedo et al., 2024). DoReMi and RegMix optimize pretraining mixtures using proxy runs (Xie et al., 2023a; Liu et al., 2024a). Recent SFT mixture work models effective transferred data to choose source weights (Li et al., 2025), while DSIR matches a selected distribution to a target distribution (Xie et al., 2023b). For instruction tuning, LESS uses gradient influence, TSDS uses target-distribution alignment, and NICE optimizes selection against a non-differentiable evaluation metric (Xia et al., 2024; Liu et al., 2024b; Wang et al., 2025). These methods answer which data to choose. We ask how the utility of a measured choice scales with model size and token budget.

Alignment metrics and distribution specificity. Domain adaptation studies learning under source–target distribution mismatch (Ben-David et al., 2010). Our score places a practical target-relative measure of that mismatch inside a fine-tuning response surface. Our score combines SBERT similarity (Reimers & Gurevych, 2019) and ZIP-FIT compression similarity (Obbad et al., 2024). Neither component is an intrinsic quality measure. The score is defined relative to a target set, so it captures target relevance and can encourage specialization. This distinction is central to our accuracy results: selection improves the aligned target while it can hurt a nearby but different distribution. We therefore use *data–task alignment* throughout to keep this target dependence explicit.

3 Problem Setting and Measurements

Let $\mathcal{P}$ be a candidate training pool and $\mathcal{D}_{\text{target}}$ a held-out target set. A scoring function $s(x) \in [0, 1]$ measures the relevance of $x \in \mathcal{P}$ to that target. For a selected subset $\mathcal{S}$, we define token-weighted alignment

$$\alpha(\mathcal{S}) = \frac{\sum_{x \in \mathcal{S}} t(x)s(x)}{\sum_{x \in \mathcal{S}} t(x)}, \tag{2}$$

where $t(x)$ is the number of training tokens. The outcome is response-token cross-entropy on $\mathcal{D}_{\text{target}}$, with prompt tokens masked. The scaling variables are parameter count N, subset token count T, and α.

For the main selector,

$$s(x) = \tfrac{1}{2}\widetilde{s}_{\text{SBERT}}(x) + \tfrac{1}{2}\widetilde{s}_{\text{ZIP}}(x), \tag{3}$$

where each component is min–max normalized over the pool. SBERT uses maximum cosine similarity to target references under `all-mpnet-base-v2`. ZIP-FIT uses compression-ratio similarity under a 128KB zstd dictionary trained on those references. We form quantile-ranked subsets at a fixed T and record their realized α. Equation (3) supplies the operational measurement used in the primary grid. We later construct an alternative with a frozen gradient proxy to test whether target-relative alignment remains useful when the selector changes.

The token weighting in Equation (2) matches the unit counted by T. It also prevents a subset containing many short, highly scored examples from appearing more aligned than the token stream seen by the optimizer. Quantile labels such as Q90 describe how a subset was constructed. The continuous realized α is the quantity used by the model. This distinction lets two subsets with different labels occupy nearby points on the same response surface and prevents the law from reducing to a comparison among named buckets.

4 Alignment as a Scaling Coordinate

The standard two-axis form

$$L_0(N, T) = a + bN^{-\beta} + cT^{-\gamma} \tag{4}$$

attributes predictable loss variation to model size and training tokens. Our first-order extension in Equation (1) assigns the remaining systematic variation to measured target alignment. The coefficient d is the local loss reduction associated with one unit of α after accounting for N and T. We

fit it globally rather than fitting a separate effect for each model size, then test that restriction in crossed-budget experiments.

The additive form has two useful interpretations. First, it treats alignment as a shift in the attainable loss surface over the measured region. Second, it converts selection gains into compute-equivalent units. Holding T fixed, the parameter multiplier that produces the same predicted loss change as moving from α_1 to α_2 is

$$\frac{N_2}{N_1} = \left(1 - \frac{d(\alpha_2 - \alpha_1)N_1^{\beta}}{b}\right)^{-1/\beta}, \tag{5}$$

when the term in parentheses is positive. This exchange rate applies within the observed range.

We compare this form against three alternatives. The no-alignment baseline in Equation (4) tests whether N and T already explain the grid. Resource multipliers let alignment change the effective amount of model or token compute. Quadratic and saturating terms test whether the alignment response is curved inside the observed range. We use BIC to penalize added flexibility and cell bootstrap to measure sensitivity. We then create new experiments for the conditions under which the models disagree. The crossed grid tests whether d varies with resources, and the extreme-quantile study tests whether an in-range fit can survive loss of pool support.

5 Experimental Design

Primary grid. The math pool contains 1.0M instruction examples from six sources and is dominated by NuminaMath. The target is a held-out 747-example set of grade-school mathematical problems. We train Qwen3 at 0.6B, 1.7B, 4B, and 8B. At $T = 21$M tokens, we use five score quantiles $\{\text{Q10}, \text{Q30}, \text{Q50}, \text{Q70}, \text{Q90}\}$ and a random baseline. At Q90 and RANDOM, we additionally vary $T \in \{2, 5, 10, 21, 42\}$M. After aggregating duplicate seed cells, this produces 51 distinct (α, N, T) observations and a $21\times$ token range. Quantile subsets are selected from the same cleaned pool, and every run uses the same instruction template and evaluation target. The intervention changes only which examples supply the fixed token budget. The evaluation set and loss definition remain fixed.

All runs use full fine-tuning in bfloat16, AdamW, cosine decay, 3% warmup, sequence length 2048, and three passes over each subset. The peak learning rate is 2×10^{-5}. We use two seeds for the 1.7B and 4B iso-token cells and selected 8B cells. The remaining primary cells use one seed due to compute. The experimental unit is a cell. The 51-cell fit therefore reflects variation across controlled training runs rather than checkpoints or individual examples. Repeated seeds estimate trajectory variability for selected cells but are not counted as additional design points.

Decontamination. Any score computed against the target can surface near-duplicates. Before selection, we remove pool examples sharing at least ten word 13-gram shingles with any target example. This removes 0.7% of the pool and reduces the fraction of Q90 examples with at least one shared shingle from 7.7% to 1.45%. Recomputing the grid after this filter changes the fitted alignment coefficient by less than 1%. The filter does not detect all semantic or template overlap, so cross-target tests remain necessary.

Models and inference. We compare a no-alignment Chinchilla form, the first-order law in Equation (1), multiplicative resource models, and curvature or interaction extensions. Fits use Huber loss and L-BFGS-B with random restarts. BIC and RMSE summarize in-sample fit. Cell bootstrap describes coefficient and model-selection sensitivity. Because N and T span a finite range, nuisance parameters summarize this design. The central object is whether adding measured alignment improves prediction and how that effect changes across controlled stress tests.

We also evaluate two held-out directions. Endpoint validation withholds the highest alignment or token level inside the original grid and tests whether the fitted response extrapolates one controlled step. The stress tests then change the construction itself. This separates interpolation quality from transfer to a new target, domain, family, selector, or selection regime.

Stress tests. We add five experiments. First, a crossed grid measures all six alignment conditions at both 5M and 21M tokens across all four Qwen3 sizes. Second, Q95, Q97, and Q99 subsets probe

extreme selection at 1.7B with matched Q90 token-budget controls. Third, an independently sampled 747-example math target is used to rescore, redecontaminate, and retrain a 16-cell grid. Fourth, a Python-filtered code pool is scored against HumanEval in two complementary grids. The original 35-cell, three-size grid combines a fixed-21M-token alignment slice with Q90/random token sweeps. A four-size fixed-21M grid enables the candidate-law comparison. Fifth, a frozen Qwen3-0.6B LoRA proxy assigns gradient-alignment scores once, after which the same ranking is used to train 0.6B, 1.7B, and 4B models. The last test makes the selector exogenous to downstream model size.

6 RESULTS

6.1 ALIGNMENT EXPLAINS LOSS LEFT OVER BY MODEL AND TOKEN SCALING

At fixed $T = 21$M, increasing α lowers held-out loss at every model size (Figure 2a). The Q90–RANDOM loss reduction is 6.7%, 9.8%, 11.2%, and 8.9% from 0.6B through 8B. The effect is therefore comparable in size to a meaningful scaling intervention and persists across the measured $13\times$ parameter range.

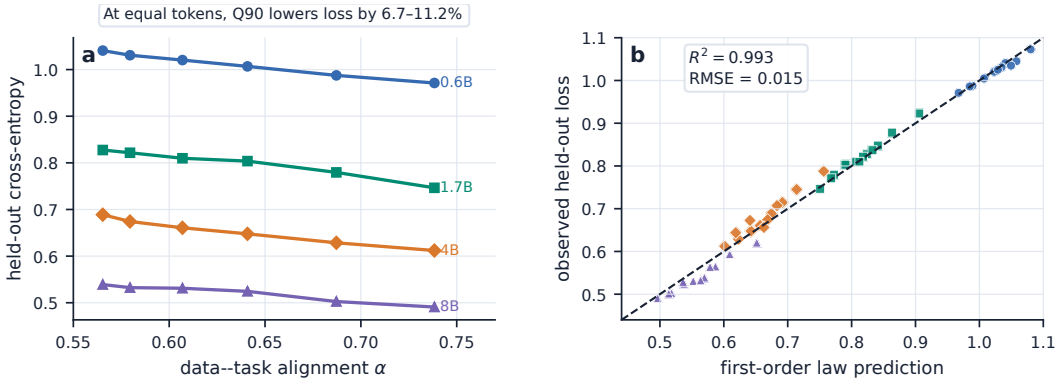

Figure 2: Alignment explains a systematic component of fine-tuning loss beyond model size and token budget. (a) At 21M tokens, held-out loss decreases with measured alignment at every Qwen3 size. (b) The first-order law predicts the full 51-cell grid with $R^2 = 0.993$ and RMSE 0.015. Each marker is one aggregated (α, N, T) cell.

Across all 51 cells, Equation (1) reaches $R^2 = 0.993$ and RMSE 0.015 (Figure 2b). The fitted alignment coefficient is $d = 0.42$, with a cell-bootstrap 95% interval of $[0.31, 0.78]$. Compared with a no-alignment model, RMSE falls from 0.035 to 0.015. Omitting α also increases the fitted model exponent from 0.185 to 0.257 and the token exponent from 0.448 to 0.508. A two-axis fit therefore absorbs composition into apparent scaling with N and T.

The effect is also useful in allocation units. Under the fitted law, moving from Q10 to Q90 has the same predicted loss reduction as increasing parameter count by approximately $1.49\times$ at 1.7B, $1.60\times$ at 4B, and $1.71\times$ at 8B, with tokens held fixed. These multipliers grow because the returns to N diminish while the fitted alignment shift remains additive. Within the observed grid, they provide a direct exchange rate between selection and scale. The support and diversity boundary studied below limits further extrapolation.

Two endpoint tests ask whether the surface predicts a held-out edge of its own design. Fitting without the highest alignment level and predicting that endpoint gives $R^2 = 0.990$. Withholding the largest token budget gives $R^2 = 0.999$. Together, these endpoint results establish one-step extrapolation within the original design. The independent target and code studies evaluate transfer beyond that design.

Table 1 compares the prespecified response surfaces on the complete 51-cell grid.

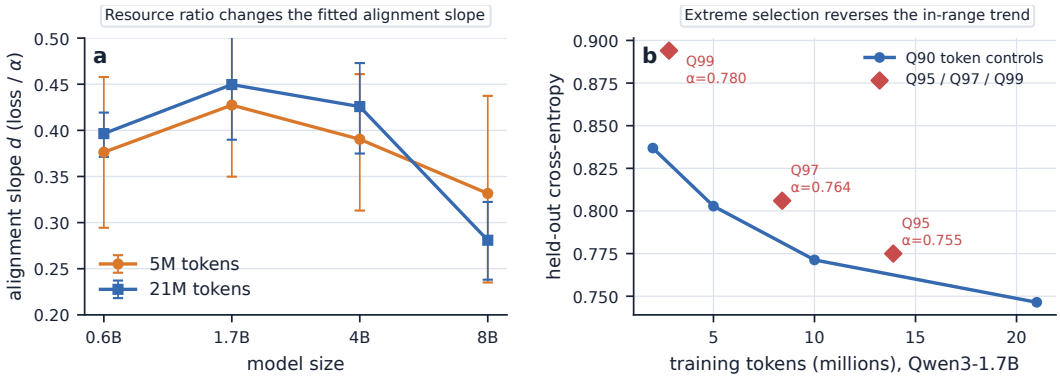

Figure 3: The first-order alignment effect varies across resource ratios and breaks under extreme selection. (a) Alignment slopes from the crossed 5M/21M design vary with the resource regime, especially at 8B; whiskers are residual-bootstrap 95% intervals over fitted cells, not training-seed intervals. (b) At 1.7B, matched Q90 controls improve as tokens increase, but Q95–Q99 depart from that curve and Q99 reverses the in-range trend. Extreme selection changes both alignment and available pool size. The matched Q90 controls isolate the small-T explanation.

Table 1: Measured alignment improves the 51-cell fit, and added curvature improves it further. BIC is lower when better. The saturating model adds one parameter relative to the first-order law. Its gain motivates the regime tests; bootstrap curvature remains uncertain.

Model	Params.	R^2	RMSE	BIC
No alignment	5	0.963	0.035	−322
Resource multiplier	6	0.988	0.020	−377
First-order alignment	6	0.993	0.015	−403
Saturating alignment	7	0.996	0.012	−427

Among the original candidates, the first-order law improves on the multiplicative resource model by 25 BIC points (Table 1). Adding curvature reveals a second pattern: a saturating alignment curve improves BIC by another 26 points. Its quadratic analogue has a bootstrap curvature interval that includes zero, and both flexible curves are estimated only inside the original range. We therefore retain Equation (1) as the minimal in-range summary and test its boundaries directly.

The comparison clarifies the modeling choice. A flexible response can describe the 51 cells slightly better, but the first-order term already removes most of the error left by the no-alignment law. We use the smaller form when translating alignment into compute-equivalent units and use the broader models to test whether the data demand a different regime.

6.2 The alignment effect depends weakly on token-to-parameter ratio

The crossed two-budget experiment separates model size from token-to-parameter ratio. Figure 3a shows per-size alignment slopes at 5M and 21M tokens. A shared-d model remains a useful approximation, but a smooth $d(T/N)$ extension improves BIC by 3.7 points. A smooth model-size-only interaction does not improve on the shared slope. The evidence supports mild resource-ratio dependence, while the variation does not follow a monotonic trend in model size.

Accordingly, the 51-cell fit supports a compact scalar correction over its measured region, with modest variation in its value across data-to-parameter ratios.

6.3 Extreme selection produces diminishing returns and reversal

Quantile selection cannot increase indefinitely without narrowing the candidate pool. At 1.7B, Q95, Q97, and Q99 have alignments 0.755, 0.764, and 0.780, but their available token counts fall to 13.9M, 8.4M, and 2.8M. Lower T alone does not explain the result. Across matched Q90 controls from 2M to 21M, the first-order prediction is accurate within 0.008 loss. The extreme cells depart increasingly from this control, and Q99 reaches loss 0.894, worse than the 2M-token Q90 control at 0.837 (Figure 3b).

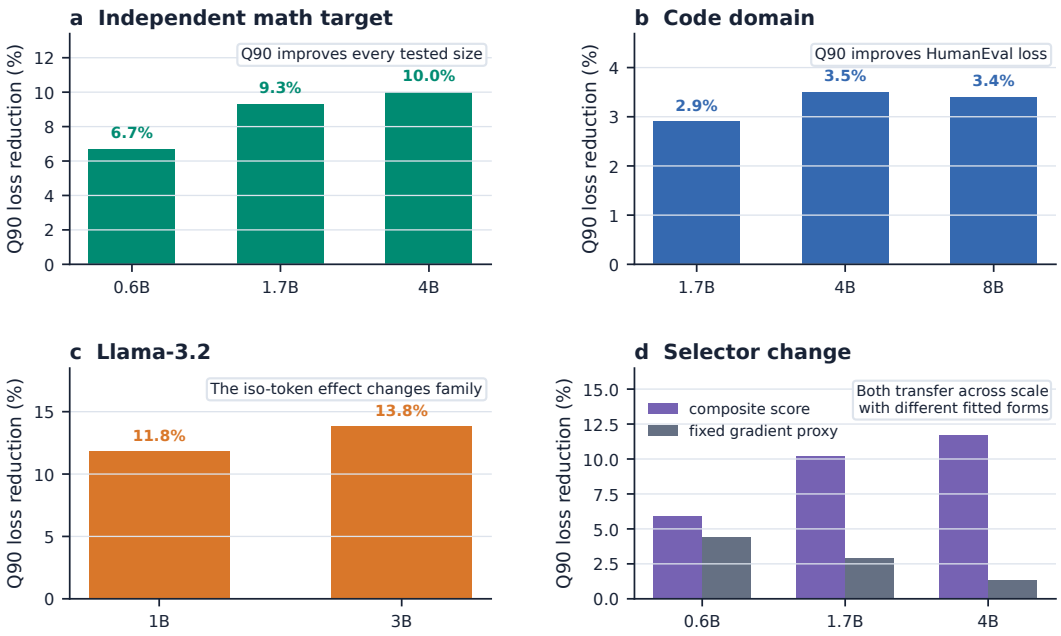

Figure 4: Alignment transfers across targets, domains, and model families, while the preferred law depends on its measurement. Each panel changes one part of the primary design. The independent target, code, and Llama studies reproduce the alignment benefit. A frozen gradient proxy also transfers across scale and favors an N-dependent coefficient, while the composite score favors a shared first-order slope. New transfer cells use one training seed, so the bars report measured contrasts without seed intervals.

The reversal marks a boundary on the selector. At extreme quantiles, support can collapse even as mean similarity rises. Because the experiment does not vary these properties independently, it identifies the boundary while leaving its cause unresolved. Refitting with the three extreme cells changes d by only 2.6%, showing that the first-order coefficient remains stable inside its original range even though extrapolation beyond that range fails.

6.4 The coordinate transfers more broadly than its exact law

The alignment effect replicates when the original target, domain, family, or selector is changed (Figure 4). On an independently sampled math target, Q90 lowers loss relative to Random by 6.7–9.3% at 0.6B, 1.7B, and 4B. A compact 16-cell refit reaches $R^2 = 0.998$, although its small size makes functional-form selection exploratory. On code, Q90 lowers HumanEval loss by 2.9–3.5% at 1.7B, 4B, and 8B. The 35-cell grid includes an iso-token alignment slice and Q90/random token sweeps. A separate 23-cell, fixed-21M grid spans 0.6B through 8B; its additive fit reaches $R^2 = 0.952$. On the leakage-free MBPP+ accuracy check, Q90 improves pass@1 by 1.9 points at 4B and changes it by -0.5 points at 8B. The loss effect is consistent while cross-benchmark accuracy is modest. On Llama-3.2, the iso-token Q90 loss reduction is 11.8% at 1B and 13.8% at 3B.

The selector change is the most diagnostic test. We train a 0.6B LoRA proxy on a disjoint 5% warmup split, score the remaining pool once using optimizer-aware gradient alignment, and freeze that ranking before training larger models. The gradient score remains predictive: adding it reduces fixed-T RMSE from 0.0133 to 0.0068. Unlike the composite score, however, it favors an N-dependent slope. On the same reduced pool, the composite selector favors the shared additive form and achieves lower Q90 loss at all three sizes. These results show that a measured alignment coordinate can transfer across scale while its coefficient and functional form remain properties of the selector and training regime.

The two selectors encode different evidence. The composite score combines semantic and distributional signals, whereas the gradient proxy measures local optimizer-aware influence under a smaller model. Both improve over the same random baseline, but the composite effect grows from 5.9% to 11.7% across 0.6B–4B while the frozen gradient-proxy gain contracts from 4.4% to 1.3%. A selector learned at one scale can remain directionally useful without preserving a scale-invariant coefficient.

6.5 Both components contribute to the composite measurement

Both components contribute to the composite score. On the primary 21M-token slice, Q90 selected by the full score yields losses of 0.747, 0.612, and 0.491 at 1.7B, 4B, and 8B. ZIP-FIT alone gives 0.861, 0.699, and 0.568, while SBERT alone gives 0.758, 0.638, and 0.501. The semantic component is stronger than compression alone, but the combined ranking remains best at every tested size. This supports a coordinate assembled from complementary semantic and distributional signals.

6.6 Alignment measures target specialization

Selection improves the distribution used to define α, while the gain changes with the evaluation target. At $T = 21$M, Q90 improves GSM8K exact match over RANDOM by $+0.7$, $+3.1$, $+3.3$, and $+0.4$ points from 0.6B to 8B. On MATH-500, the same models change by -1.8, -2.6, -9.4, and -1.6 points. The target scores reflect this ordering: the composite alignment of GSM8K to the development target is 0.709, compared with 0.404 for MATH-500. The result is target-specific specialization: the selected examples are better matched to the measured target, and their advantage is specific to that target (Figure 5).

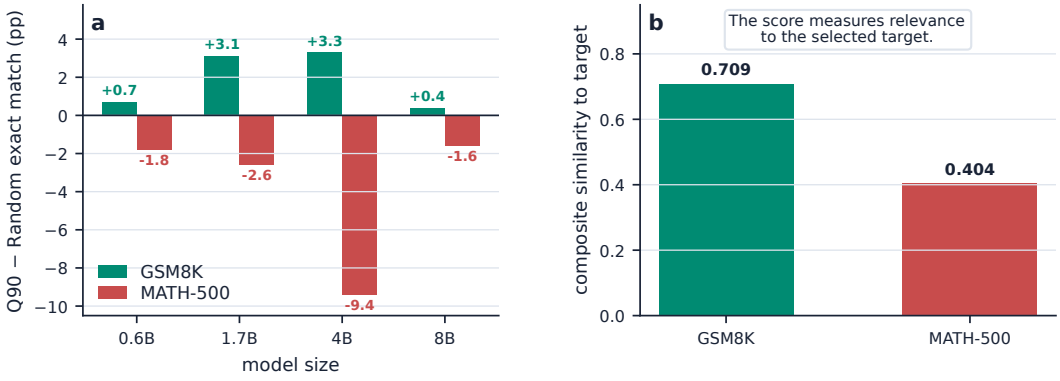

Figure 5: Data–task alignment predicts target-specific specialization. (a) Q90 selection improves exact match on GSM8K but can reduce MATH-500 accuracy under the same training contract. (b) The composite score places GSM8K substantially closer to the development target. The direction of transfer follows measured target relevance.

This finding also explains why the scalar should enter a scaling analysis only with its target attached. α explicitly denotes $\alpha(\mathcal{S}, \mathcal{D}_{\text{target}}, s)$, so its value changes with the target and scoring rule.

7 Discussion

7.1 From data selection to predictable allocation

Data composition is a scaling variable and an otherwise unmodeled source of variation. In the primary grid, alignment explains loss that a model in N and T would otherwise absorb into its compute exponents. This suggests a practical workflow: measure target relevance before training, estimate its value inside a validated regime, and compare the expected gain with the cost of additional parameters or tokens. The resulting allocation problem is richer than choosing the largest dataset available.

The compute-equivalent view makes this point concrete. Inside the primary grid, moving from Q10 to Q90 is comparable to a $1.49\times$–$1.71\times$ increase in parameter count, depending on the starting scale. Realizing that gain still requires a sufficiently broad pool, a representative target, and a score whose ranking remains meaningful under the downstream trainer. The result places data choice and compute choice in the same planning problem. A practitioner who measures only N and T can spend more while moving along the wrong response surface.

Our experiments support a simple regime map. Alignment is most useful when the target is defined, the candidate pool retains support, and the score is validated near the intended resource ratio. It becomes less predictable when selection removes most of the pool or when the selector is transferred

Table 2: The evidence separates a transferable coordinate from the conditions that determine its law. The direction of the alignment effect is stable across several changes, while its magnitude and functional form are regime-dependent.

Intervention	What transfers	What changes
Primary 51-cell grid	Higher α predicts lower loss	Flexible curves improve in-range fit
Crossed budgets	Shared slope remains useful	Mild dependence on T/N
Q95–Q99 selection	Target relevance remains high	Narrow support causes reversal
Target, domain, family	Q90 improves held-out loss	Effect size and available axes differ
Frozen gradient proxy	Alignment remains predictive	N-dependent slope is preferred

across optimization regimes. A useful account therefore pairs the compact law with the interventions that reveal its boundary.

7.2 Scope of the alignment coordinate

The experiments also define the scalar's limits. The first-order law is accurate within the controlled composite-score grid, but the value of alignment changes mildly with the resource ratio. Extreme selection reverses the trend, and a gradient-based selector prefers a different interaction with scale. A single scalar necessarily compresses target relevance, diversity, support, and optimization geometry. Its value comes from being measurable and predictive over a stated region while leaving these other properties available for separate analysis.

Across the independent target, code, and Llama studies, the direction of the alignment effect recurs beyond one evaluation set and one dense model family. The extreme-quantile reversal separates mean target relevance from coverage. The frozen gradient-proxy study shows that even a selector with direct optimizer information can change its scaling behavior across model size. Together, these results favor measured, target-relative response surfaces over a single global ordering of examples.

For scaling-law methodology, the omitted-variable result is as important as the fitted coefficient. Without alignment, model and token exponents inherit variation caused by composition. Two studies can therefore estimate different compute returns because training relevance moves with scale or budget. A target-relative coordinate makes that assumption visible and testable.

7.3 Limitations and conclusion

Our evidence spans two dense model families and two domains under one main fine-tuning recipe, with few seeds. Code and Llama do not span a full token axis. Extreme selection changes pool size with alignment, though matched-token controls reject the simplest small-T explanation. Cross-entropy is the stable primary outcome; accuracy gains are smaller and distribution-specific. More selectors, targets, architectures, and budgets are needed before treating any coefficient as universal.

The score also depends on the target sample. A small or unrepresentative target can precisely rank the wrong distribution. Independent-target, decontamination, and distribution-specific audits reduce but do not remove this concern. Future work should model target uncertainty and vary alignment, diversity, and support independently to test whether the extreme-selection boundary is predictable.

Across this controlled scope, equal token budgets provide different learning opportunities. Data–task alignment supplies a third coordinate that predicts a large component of held-out loss beyond N and T. Its boundary is equally informative: target relevance helps within a regime, then loses value when selection narrows the experience too far. Predictable fine-tuning therefore depends on both how much data a model sees and how well that data covers the target it is expected to learn.

AI USE STATEMENT

GPT-5.6 Sol was used to assist with debugging infrastructure, editing for concision, and figure-generation scripting. The authors reviewed all AI-assisted work and take responsibility for the final paper.

REPRODUCIBILITY STATEMENT

Appendix A records datasets, subset construction, training settings, fit procedures, and the new stress-test grids. The accompanying anonymous supplement provides alignment-scoring, subset-construction, decontamination, fitting, and evaluation code together with configurations, subset manifests, and machine-readable results.

REFERENCES

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79: 151–175, 2010. doi: 10.1007/s10994-009-5152-4. URL https://doi.org/10.1007/s10994-009-5152-4.

Ethan Caballero, Kshitij Gupta, Irina Rish, and David Krueger. Broken neural scaling laws. *arXiv preprint arXiv:2210.14891*, 2022. URL https://arxiv.org/abs/2210.14891.

Tatsunori Hashimoto. Model performance scaling with multiple data sources. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4107–4116. PMLR, 2021. URL http://proceedings.mlr.press/v139/hashimoto21a.html.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017. URL https://arxiv.org/abs/1712.00409.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. URL https://arxiv.org/abs/2203.15556.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. URL https://arxiv.org/abs/2001.08361.

Yuan Li, Zhengzhong Liu, and Eric Xing. Data mixing optimization for supervised fine-tuning of large language models. *arXiv preprint arXiv:2508.11953*, 2025. URL https://arxiv.org/abs/2508.11953.

Haowei Lin, Baizhou Huang, Haotian Ye, Qinyu Chen, Zihao Wang, Sujian Li, Jianzhu Ma, Xiaojun Wan, James Zou, and Yitao Liang. Selecting large language model to fine-tune via rectified scaling law. *arXiv preprint arXiv:2402.02314*, 2024. URL https://arxiv.org/abs/2402.02314.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. Regmix: Data mixture as regression for language model pre-training. *arXiv preprint arXiv:2407.01492*, 2024a. URL https://arxiv.org/abs/2407.01492.

Zifan Liu, Amin Karbasi, and Theodoros Rekatsinas. TSDS: Data selection for task-specific model finetuning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 10117–10147, 2024b. URL https://arxiv.org/abs/2410.11303.

Elyas Obbad, Iddah Mlauzi, Brando Miranda, Rylan Schaeffer, Kamal Obbad, Suhana Bedi, and Sanmi Koyejo. Zip-fit: Embedding-free data selection via compression-based alignment. *arXiv preprint arXiv:2410.18194*, 2024. URL https://arxiv.org/abs/2410.18194.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*, 2024. URL https://arxiv.org/abs/2406.17557.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019. URL https://arxiv.org/abs/1908.10084.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: Beating power law scaling via data pruning. *arXiv preprint arXiv:2206.14486*, 2022. URL https://arxiv.org/abs/2206.14486.

Jingtan Wang, Xiaoqiang Lin, Rui Qiao, Pang Wei Koh, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. NICE data selection for instruction tuning in LLMs with non-differentiable evaluation metric. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63662–63689, 2025.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2402.04333.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. In *Advances in Neural Information Processing Systems*, 2023a. URL https://arxiv.org/abs/2305.10429.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34201–34227, 2023b. URL https://arxiv.org/abs/2302.03169.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/cc84bfabe6389d8883fc2071c848f62a-Abstract-Conference.html.

Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. When scaling meets LLM finetuning: The effect of data, model and finetuning method. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2402.17193.

A Reproducibility Details

Model and optimizer settings. All primary Qwen3 runs use full fine-tuning in bfloat16, AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0.01, peak learning rate 2×10^{-5}, cosine decay, 3% warmup, sequence length 2048, gradient checkpointing, and per-device batch size 8. Each model trains for three passes over its iso-token subset. The primary evaluation uses response-token cross-entropy with prompt tokens masked and the training instruction template.

Primary data. The 1.0M-example math pool is 85.7% NuminaMath, with OrcaMath, MathInstruct, MetaMath, OpenWebMath, and AQuA-RAT forming most of the remainder. The code pool contains 617K Python-filtered examples: 81.0% OpenCodeInstruct, 15.0% CodeFeedback, and 4.0% Evol-Instruct after exact-string deduplication.

Fit procedure. Each form is fit with Huber loss and L-BFGS-B from at least 200 random restarts. BIC uses Gaussian per-cell residuals. The primary confidence interval for d uses 200 cell-level bootstrap resamples and therefore summarizes cell sampling. Most primary cells have one seed, while selected iso-token cells have two, so the interval is not a detailed model of training-seed variance.

B Candidate Functional Forms

The original comparison includes:

$$L_0 = a + bN^{-\beta} + cT^{-\gamma}, \tag{6}$$

$$L_{\text{lin}} = L_0 - d\alpha, \tag{7}$$

$$L_{N\times\alpha} = a + b(N\alpha^{\lambda_N})^{-\beta} + cT^{-\gamma}, \tag{8}$$

$$L_{\text{res}} = a + b(N\alpha^{\lambda})^{-\beta} + c(T\alpha^{\lambda})^{-\gamma}. \tag{9}$$

We additionally test $L_0 - d_1\alpha - d_2\alpha^2$ and $L_0 - d(1 - e^{-k\alpha})$. The latter has the best in-sample BIC in Table 1. Its role is diagnostic: with only one observed alignment range, the observed curvature does not determine the response outside the grid.

C Crossed-Budget Model Comparison

The crossed design contains 48 cells: six alignment conditions, two token budgets, and four model sizes. We compare a shared d, free d_N, smooth $d(N)$, and smooth $d(T/N)$. The shared model has RMSE 0.0169 and BIC -368.7. Smooth $d(N)$ nests the shared form at zero interaction and loses by 3.9 BIC points. Smooth $d(T/N)$ reaches RMSE 0.0156 and BIC -372.5, a 3.7 point improvement. With only two token columns, we do not fit a full $d(N, T/N)$ surface.

D Independent-Target and Code Fits

Table 3: Independent-target Q90–Random loss changes. The target examples, scores, decontamination filter, and trained subsets are disjoint from the primary target experiment.

Model	Q90 loss	Random loss	Relative change
Qwen3-0.6B	0.960	1.029	-6.7%
Qwen3-1.7B	0.741	0.816	-9.3%
Qwen3-4B	0.605	0.673	-10.0%

The independent-target grid has 16 cells. Its additive fit has $R^2 = 0.998$, RMSE 0.0058, and BIC -148.0, compared with -141.5 for the resource multiplier. Because the grid is compact and single-seed, we use it primarily as a replication of the alignment effect.

The original code grid contains 35 cells over Qwen3-1.7B, 4B, and 8B. It combines a 21M-token alignment slice with Q90/random token sweeps from 2M to 42M and supplies the transfer contrasts in Figure 4. A separate 23-cell candidate-law grid holds $T = 21$M fixed and spans Qwen3-0.6B, 1.7B, 4B, and 8B with five or six alignment conditions per size. On that grid, the additive model reaches $R^2 = 0.952$, RMSE 0.0588, and BIC -111.5, compared with $R^2 = 0.938$ and BIC -108.7 without alignment.

E Selector Ablations

On a shared reduced pool, the composite-score selector and fixed gradient proxy use the same random baseline and $T = 21$M. The composite Q90 losses are 0.973, 0.748, and 0.609 at 0.6B, 1.7B, and 4B. Gradient Q90 gives 0.988, 0.809, and 0.681. The shared random losses are 1.034, 0.833, and 0.690. The gradient score still improves over random at every size, but its per-size slopes decrease from 0.356 to 0.205 to 0.073. An N-dependent slope is therefore preferred by BIC.

F Accuracy and Distribution Specificity

Composite similarity to the development target is 0.709 for GSM8K and 0.404 for MATH-500. The corresponding SBERT scores are 0.719 and 0.441. ZIP-FIT scores are 0.522 and 0.275. These measurements show that the selector captures distribution-specific relevance.

Table 4: Exact-match accuracy at $T = 21$M. Parentheses show Q90 minus Random in percentage points. GSM8K is closer to the alignment target than MATH-500.

	0.6B	1.7B	4B	8B
GSM8K Q90	48.4 (+0.7)	66.1 (+3.1)	79.6 (+3.3)	74.3 (+0.4)
MATH-500 Q90	23.6 (−1.8)	38.8 (−2.6)	41.8 (−9.4)	49.2 (−1.6)

G Decontamination Robustness

Because the selector is defined relative to the target, leakage control is part of the measurement pipeline. The fixed ten-shingle filter removes 0.7% of the pool and reduces top-quantile target overlap from 7.68% to 1.45% (Table 5). Refitting before and after this intervention changes d by less than 1%, so the main alignment coefficient is not explained by the removed near-duplicates.

Table 5: Target 13-gram overlap before and after the fixed decontamination filter. Values are the percentage of each subset sharing at least one shingle with the development target.

Subset	Before filter (%)	After filter (%)
Q10	1.03	0.23
Q30	1.34	0.30
Q50	1.92	0.38
Q70	3.09	0.30
Q90	7.68	1.45
Random	0.88	0.18

H Exploratory 14B Observation

We trained Qwen3-14B Q90 and Random cells at $T = 21$M with seeds 41 and 42, outside the fitted grid. Mean held-out loss is 0.4510 for Q90 and 0.4584 for Random, a 1.6% reduction that is substantially smaller than the 8.9% reduction at 8B. GSM8K exact match improves by 1.6 points on average, while MATH-500 improves by 3.7 points, reversing the negative in-range MATH-500 contrast. These observations are exploratory and are not used to fit the reported law.

Table 6: Qwen3-14B iso-token results at $T = 21$M. The cells are outside the fitted 0.6B–8B grid.

Metric	Condition	Seed 41	Seed 42	Mean
Loss	Q90	0.4504	0.4516	0.4510
Loss	Random	0.4586	0.4582	0.4584
GSM8K EM (%)	Q90	81.4	81.1	81.2
GSM8K EM (%)	Random	80.8	78.5	79.6
MATH-500 EM (%)	Q90	55.4	58.4	56.9
MATH-500 EM (%)	Random	53.8	52.6	53.2

I Compute and Assets

The primary training pipeline ran through Modal, with a small number of legacy and smoke jobs on two on-prem clusters. The archived original release reports approximately 400 GPU-hours on NVIDIA H200 and B200 GPUs. A legacy on-prem 8B run was retrained on Modal B200 so the fitted grid shares one software and hardware path. The added crossed-budget, independent-target, and selector training manifests identify Modal B200 workers. Fixed-proxy scoring and smoke tests also used an on-prem B200. Each Modal training job occupied one GPU. Reference wall times were approximately 25, 75, and 150 minutes for 1.7B, 4B, and 8B runs, respectively.

The complete study contains the 51-cell primary grid, a 48-cell crossed grid, three extreme-quantile runs, a 16-cell independent-target grid, 35-cell and 23-cell code grids, six Llama replication runs, and matched 12-cell composite and fixed-proxy selector grids. Qwen3 is used under Apache 2.0 and Llama-3.2 under the Llama 3.2 Community License. The released code is MIT licensed. The supplement records source identifiers for the composite datasets and evaluation benchmarks. Their upstream terms remain in force, and no dataset contents or model weights are redistributed.