MIX, DON'T TUNE: BILINGUAL PRE-TRAINING OUTPERFORMS HYPERPARAMETER SEARCH IN DATA-CONSTRAINED SETTINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

For most languages of the world, language model pre-training operates in a data-constrained regime where models must repeat their training data many times, degrading generalization. Two remedies exist: aggressive hyperparameter tuning such as high weight decay, and mixing in data from a high-resource auxiliary language to directly aid the low-resource target. While hyperparameter tuning regularizes the model by shrinking weights to restrict network capacity, auxiliary data mixing uses a tunable mixing ratio to expand the training distribution and diversify the training signal with new knowledge. Both offer a principled way to improve training in a data-constrained domain. We compare these levers systematically across four model scales from 150M to 1.43B parameters, using Arabic as the low-resource target and English as the auxiliary, over approximately 1000 pre-training runs. Three findings emerge. First, mixing yields larger improvements than hyperparameter tuning on both validation loss and downstream task accuracy, and the gap grows with model size. Second, we quantify how much mixing helps: it boosts performance by an amount equivalent to 2–3$\times$ the unique target data on validation loss and 2–13$\times$ on downstream task accuracy, with the gain scaling steeply with model size. Third, this divergence indicates that target-language validation loss underestimates mixing's value on downstream evaluation. Mixing regularizes by diversifying the training signal and contributes knowledge the repeated target corpus cannot supply; validation loss captures only the first effect. Our practical recommendations are: mix in a high-resource language, prioritize the mixing ratio over hyperparameter tuning, and transfer hyperparameters from a small proxy model via μP.

1 INTRODUCTION

As compute budgets grow faster than available text, language model pre-training increasingly operates in a data-constrained regime where models must repeat their training data many times (Muennighoff et al., 2025). Beyond a threshold, repetition actively damages model quality, disproportionately degrading generalization while shifting capacity toward memorization, with larger models hitting the degradation region at fewer repeated epochs (Hernandez et al., 2022). Data scarcity is especially acute for low-resource languages (Penedo et al., 2025), but arises whenever a high-value corpus is too small to fill the compute budget. Prior work proposes two remedies: Kim et al. (2025) show that under extreme repetition, naive scaling produces a U-shaped loss curve as model size grows, but aggressive regularization (weight decay up to 30$\times$ standard practice) restores monotonically decreasing scaling, while Seto et al. (2025) show that mixing in data from a high-resource auxiliary language can boost target-language performance without explicitly modifying the model or objective.

These results suggest two levers for a practitioner training on scarce data: regularize harder to get more out of what you have, or bring in auxiliary data to diversify the training signal. The levers are not exclusive, and one can mix and tune at once. Each, however, carries its own search cost: tuning needs a hyperparameter grid at every model scale, and mixing needs a search over the mixing ratio. Whether both searches are worth running depends on what each lever contributes alone and on top of the other. The two mechanisms are also not equivalent. Regularization curbs overfitting to the repeated target corpus; auxiliary data also adds capabilities the target corpus cannot teach.

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.

Validation loss, computed on a held-out sample of the same scarce target distribution, directly reflects the first effect. Throughout, we use "regularization" in this strict sense: an intervention that reduces overfitting to the target corpus, measured by target-language validation loss. Because proxy metrics can fail to track the behaviors practitioners care about (Theis et al., 2016; Huszár, 2015; Nalisnick et al., 2019), comparing the two remedies demands evaluating them on both validation loss and downstream performance. We do this across four model scales (150M–1.43B parameters), using Arabic subsampled to 200M unique tokens as a controlled stand-in for a data-constrained target, with English as the auxiliary, and ask three questions: (i) when pre-training with limited target-language data, which lever improves the model more, hyperparameter (HP) tuning or mixing in a high-resource language? (ii) to what extent can language mixing make up for the lack of target-language data? (iii) does target-language validation loss account for downstream performance, and if not, in what way? We find that (i) mixing improves Arabic validation loss more than HP tuning, with a gap that grows with model size (12.9% against 3.1% for the 1.43B-parameter model), while tuning on top of mixing adds under 1%; (ii) mixing is worth the equivalent of 2–3$\times$ more unique Arabic tokens on validation loss and 2–13$\times$ on ARC Easy; and (iii) validation loss can undercount mixing's value, crediting its regularization effect but not the knowledge English adds.

2 Related work

Data-constrained scaling laws. Muennighoff et al. (2025) generalize Chinchilla scaling laws to the data-constrained regime, showing that repeated data follows an exponential saturation curve with diminishing returns beyond roughly 4 epochs. Hernandez et al. (2022) study partial repetition and find that it damages induction heads, shifting capacity from generalization to memorization. Kim et al. (2025) restore monotone scaling under extreme repetition with aggressive regularization (§1). Gadre et al. (2024) show that scaling laws extrapolate reliably in the over-training regime and can predict downstream performance. All four studies are English-only and do not consider cross-lingual data augmentation as a strategy for data efficiency. We add this dimension: bilingual mixing as a lever alongside regularization, and we quantify their relative contributions via variance decomposition.

Bilingual and multilingual pre-training. Multilingual scaling laws have been studied at the level of language families (He et al., 2024), in the curse-of-multilinguality regime across 250 languages (Chang et al., 2023), and via adaptive transfer laws spanning hundreds of languages (Longpre et al., 2026). These studies target compute-optimal sampling across many languages rather than the data-constrained bilingual regime. Seto et al. (2025) train bilingual models from scratch with a data-constrained target language (German, 250M tokens) and English auxiliary, sweep several mixing ratios at 1B and 3B, and discuss how auxiliary data substitutes for fresh target data. Broader multilingual pre-training (Lin et al., 2022; Conneau et al., 2020) targets language coverage rather than controlled mixing analysis, and the continued pre-training approach of Lin et al. (2024) cannot isolate the value of bilingual mixing because it starts from a strong English model. We complement this line by decomposing variance against HP tuning, sweeping four model scales from 150M to 1.43B, formalizing the data multiplier, and identifying a divergence between validation loss and benchmark performance.

Data mixture optimization. A growing literature optimizes domain mixture weights for English pre-training: DoReMi (Xie et al., 2023) uses distributionally robust optimization, RegMix (Liu et al., 2025) uses regression from small proxy models, Ye et al. (2025) derive parametric mixing laws predicting loss as a function of domain proportions, and Shukor et al. (2025) derive scaling laws for optimal data mixtures that extrapolate across scales and to unseen domain weights. These methods operate in a compute-constrained, many-domain setting and optimize the mixture weights. Closest to our setting, concurrent work by Sedova et al. (2026) derives a repetition-aware scaling law for mixture pre-training under data constraints, predicting target-domain loss from model size, compute, target-pool size, and mixture weight (App. I). A complementary line of work studies dynamic schedules in which mixture proportions evolve during training (Liang et al., 2025; Dong et al., 2024), an axis orthogonal to the static-ratio analysis we conduct here. We ask a different question: in a data-constrained, bilingual setting with a single mixing parameter, how much value does mixing provide compared to hyperparameter tuning, and how much unique target data is it worth?

Hyperparameter sensitivity analysis. Decomposing performance variance into per-factor contributions is the standard tool for hyperparameter importance (Hutter et al., 2014; Probst et al., 2018;

Watanabe et al., 2023). The LLM literature has instead used scaling laws (Shukor et al., 2025; Gadre et al., 2024) and regression surrogates over mixture weights (Liu et al., 2025; Ye et al., 2025), which predict the mean response surface but do not partition variance into main and interaction effects. We apply variance decomposition (classical ANOVA, since our grid is exhaustively trained) to ask whether a suboptimal choice of data mixing or hyperparameters costs more, and how the answer shifts with model scale.

3 EXPERIMENTAL SETUP

Data. We use Arabic as the low-resource (LR) target language and English as the high-resource (HR) auxiliary. The Arabic corpus is a subset of FineWeb2 (Penedo et al., 2025), which contains approximately 55B Arabic tokens. We subsample to $D_{\text{LR}} = 200\text{M}$ unique tokens, following the data-constrained regime of Kim et al. (2025). Arabic is therefore not low-resource in absolute terms. We use the term operationally: a target language whose unique-token budget is small relative to the compute budget. We create that condition by subsampling rather than by selecting an already-scarce language. Abundance is what makes the design possible, since stating that mixing is worth $N\times$ more unique Arabic data (§4.4) means training monolingual baselines on corpora of up to 2B unique tokens, which a genuinely scarce language could not supply. The validation set is a disjoint subset of FineWeb2. The English corpus comes from FineWeb (Penedo et al., 2024) and is effectively unlimited at our training scales (approximately 350B tokens). In the monolingual setting, models train on Arabic only; we fix the maximum number of times the Arabic data is repeated during training to $R_{\text{max}} = 100$, so every model sees $100 \times D_{\text{LR}}$ tokens regardless of scale. In the bilingual setting, each sample in a batch is drawn from Arabic with probability α and from English with probability $1 - \alpha$. We fix the total training budget to $D = 100 \times N_{\text{nonemb}}$ tokens, where N_{nonemb} is the number of non-embedding parameters (2.3B, 12.2B, 28.5B, and 97.3B tokens for the 150M, 380M, 600M, and 1.43B models, respectively). This budget splits into an LR portion $\alpha D = R_{\text{max}}\, D_{\text{LR}}$ (Arabic repeated R_{max} times) and an HR portion $(1 - \alpha)D = D_{\text{HR}}$ (fresh English), so $\alpha = R_{\text{max}}\, D_{\text{LR}}/D$ ties the per-batch mixing fraction to the Arabic repetition budget R_{max}. We sweep $R_{\text{max}} \in \{1, 2, 5, 10, 20, 50, 100\}$. To quantify how much unique Arabic data bilingual mixing is equivalent to (§4.4), we also train monolingual baselines at varying corpus sizes, subsampling FineWeb2 to create Arabic corpora of $D_{\text{LR}} \in \{25\text{M}, 50\text{M}, 100\text{M}, 200\text{M}, 500\text{M}, 1\text{B}, 2\text{B}\}$ tokens.

Compute cost of mixing. We study the data-constrained regime where compute outruns unique target-language data (Muennighoff et al., 2025; Kim et al., 2025): the scarce resource is Arabic, not GPU hours, so our comparison is between repeating Arabic under heavy regularization and mixing in English on a fixed Arabic budget.

Models. We train LLaMA-style decoder-only transformers at four scales (150M, 380M, 600M, 1.43B parameters) with the XGLM tokenizer (Lin et al., 2022) and sequence length 2048. Full architecture details are in App. B. Throughout, bare sizes ("at 1.43B") denote model parameters; data quantities are always stated in tokens.

Hyperparameters. We optimize with AdamW using default β-values. We tune two hyperparameters, weight decay (λ) and learning rate (η), over the grid $\lambda \in \{0.01, 0.1, 0.3, 1.0, 3.0\}$ and $\eta \in \{10^{-4}, 3{\times}10^{-4}, 10^{-3}, 3{\times}10^{-3}, 10^{-2}\}$. We use Maximal Update Parametrization (μP) (Yang et al., 2022) to tune once at a cheap proxy and transfer the base pair to larger scales. Concretely, a base (λ, η) chosen at proxy width $d_{\text{base}} = 512$ (our 150M model) is rescaled at target width d by $m_N = d/d_{\text{base}}$ as $\eta_{\mu\text{P}} = \eta/m_N$ and $\lambda_{\mu\text{P}} = \lambda \cdot m_N$ on hidden-layer parameters (App. B). For example, transferring $(\lambda = 0.1, \eta = 0.01)$ from the 150M proxy to the 1.43B model ($d = 2048$, $m_N = 4$) yields effective values $\lambda_{\mu\text{P}} = 0.4$ and $\eta_{\mu\text{P}} = 0.0025$. Every grid cell below refers to a base (λ, η) that μP rescales at the target scale. We distinguish two HP regimes. *Basic HP*: we run the 5×5 grid only on the 150M proxy, pick the best base (λ, η) by Arabic validation loss ($\lambda = 0.1$, $\eta = 0.01$), and apply μP to transfer those values to every larger scale. *Tuned HP*: we run the μP-scaled 5×5 grid *at each model scale* and pick the best base (λ, η) per scale; for example, bilingual training selects (λ=0.01, η=0.01) at 380M and (λ=0.1, η=0.003) at 1.43B (full per-paradigm values in Table 3, App. C). In the bilingual setting, the per-scale grid expands to $5 \times 5 \times 7$ to also sweep $R_{\text{max}} \in \{1, 2, 5, 10, 20, 50, 100\}$. Tuned HP represents an upper bound on what HP tuning can achieve at each scale.

Four paradigms. We traverse data strategies (monolingual vs. bilingual) and HP regimes (basic vs. tuned) to obtain four experimental paradigms (Table 1).

Table 1: Four experimental paradigms. R (Arabic repetitions at a checkpoint) is read along every run via the constant learning rate, so it is implicit in all paradigms; the "Sweeps" column lists only the design axes varied across runs.

Paradigm	Data	Hyperparameters	Sweeps
Monolingual, basic HP	Arabic only	μP from 150M	—
Monolingual, tuned HP	Arabic only	grid search per scale	λ, η
Bilingual, basic HP	Arabic + English	μP from 150M	$R_{\max}$
Bilingual, tuned HP	Arabic + English	grid search per scale	$R_{\max}$, λ, η

Evaluation. Our primary metric is Arabic validation loss (VL). Each checkpoint corresponds to a different repetition count R, yielding a full curve from a single run. For downstream evaluation we need benchmarks with above-random signal at our scales, which excludes MMLU (Hendrycks et al., 2021) (App. J). We use three: ARC Easy (Clark et al., 2018), SciQ (Welbl et al., 2017), and PIQA (Bisk et al., 2019). ARC Easy is evaluated on every checkpoint of every run and is the benchmark in all main-text figures. SciQ and PIQA are evaluated on the runs behind the data-multiplier analysis and replicate it (§4.4, App. D.4). We translate all three from English to Arabic with NLLB-200-3.3B (Team et al., 2022). A native Arabic speaker spot-checked 50 randomly sampled ARC Easy items and judged the translations faithful and the correct answer preserved. The splits are still machine-translated and carry English-aligned artifacts; App. J discusses what this implies for our downstream claims.

4 Results

4.1 Data mixing dominates hyperparameter tuning

We ask whether data mixing or hyperparameter tuning contributes more to Arabic language modeling performance. Our four paradigms (Table 1) vary each factor independently, letting us isolate their effects on Arabic validation loss across scales. Our experiments provide strong empirical evidence that **data mixing is the dominant lever, and its advantage over HP tuning grows with model size** (Fig. 1).

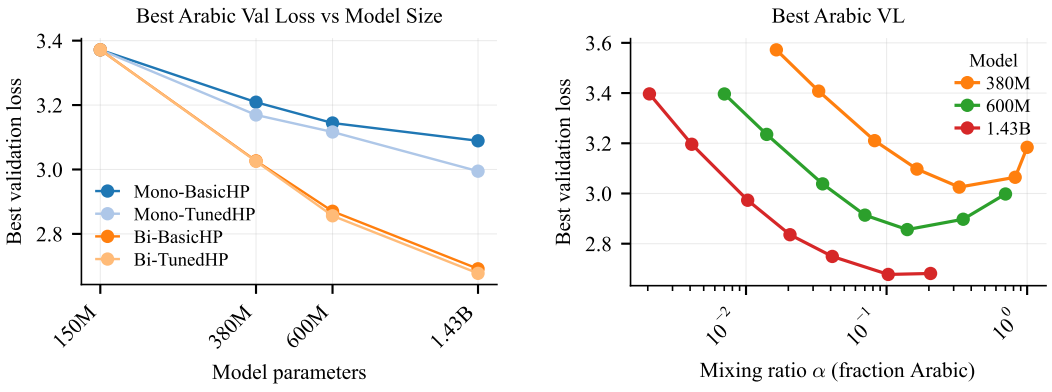


Figure 1: **Left**: best Arabic validation loss vs. model size (blue: monolingual, orange: bilingual; dark: basic HP, light: tuned HP). Bilingual training separates sharply from monolingual as scale increases. **Right**: best Arabic VL vs. mixing ratio α (fraction Arabic) for each model scale, minimized over HPs and repetitions at each (N, α). The optimum shifts toward more English as scale grows ($\alpha^* = 0.33, 0.14, 0.10$ from 380M to 1.43B), and the basin visibly flattens at 1.43B around $\alpha \in [0.1, 0.2]$.

At 150M, all four paradigms collapse to the same run: basic and tuned HP coincide by construction, since basic HP is the optimum of the 5×5 grid at the 150M proxy; and the 200M Arabic tokens

suffice without overfitting, so the bilingual paradigms select $\alpha = 1$ (all Arabic). As models grow, mixing pulls sharply ahead. At 380M, mixing improves Arabic VL by 5.7% over the monolingual baseline while HP tuning alone yields 1.2%; at 600M, 8.7% vs. 0.9%; at 1.43B, 12.9% vs. 3.1%. HP tuning on top of mixing is negligible at every scale (<1%). In concrete terms, a bilingual 600M model with μP HPs (VL 2.870) outperforms a monolingual 1.43B model with per-scale tuned HPs (VL 2.995), despite being 2.4× smaller.

Two mechanisms contribute. English data acts as a regularizer, since larger models overfit more aggressively on repeated Arabic and mixing counteracts this directly; it also supplies content the Arabic corpus does not contain. Arabic validation loss registers only the first effect, and §4.4 measures how much of mixing's value that hides. The optimal Arabic fraction reflects this, declining from $\alpha = 0.33$ at 380M to 0.14 at 600M and 0.10 at 1.43B under tuned HPs (Fig. 1, right). Once the model sees diverse data, there is little left for λ and η to improve.

This comparison holds the Arabic budget fixed, so mixing adds tokens that hyperparameter tuning cannot. §4.4 therefore repeats it against monolingual models trained on up to 2B unique Arabic tokens, a baseline on which mixing does not win automatically. The next section asks whether this ranking holds on downstream benchmarks.

4.2 Benchmark performance across paradigms

We evaluate all checkpoints on ARC Easy (Clark et al., 2018) in Arabic and English at 5-shot, selected as a knowledge-based benchmark that satisfies the four FineWeb reliability criteria (Penedo et al., 2024) (low seed variance, monotonic training curves, above-random signal, and discrimination between corpora of known quality) at small ablation scale. Figure 2 reports accuracy under several selection rules: at the best Arabic VL checkpoint (solid), at the best Arabic accuracy checkpoint (English only, dash-dot), and at peak per-language accuracy (dashed left, shaded right). Per-language splits and full per-paradigm tables are in App. C.3.

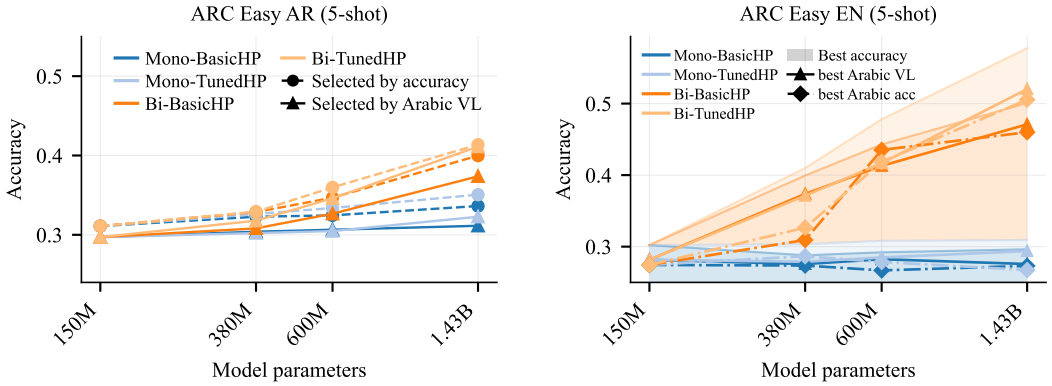


Figure 2: ARC Easy accuracy (5-shot) vs. model size. **Left** (Arabic): solid = at best Arabic VL; dashed = peak Arabic (oracle). **Right** (English): solid = at best Arabic VL; dash-dot = at best Arabic accuracy; shaded = peak English (oracle). Blue: monolingual; orange: bilingual (dark = basic HP, light = tuned HP). Bilingual scales steeply on both languages, regardless of selection criterion.

Monolingual benchmark accuracy does not scale with model size. Despite the validation loss improvements observed in Section 4.1, monolingual benchmark accuracy is essentially flat as model size grows. Arabic accuracy rises from 29.7% to only 31.1% (basic HP) or 32.3% (tuned HP) across a 10× increase in parameters from 150M to 1.43B, all near the 25% random baseline. English accuracy remains at random (∼28%) at every scale. Even with per-scale HP tuning, including weight decay up to 30× standard practice (Kim et al., 2025), the validation loss gains from aggressive regularization do not carry over to benchmarks.

Mixing scales steeply on both languages. Bilingual training with basic HPs (μP HPs from 150M) outperforms monolingual training with the same HPs by a margin that grows with scale: on Arabic ARC Easy, +0.4% at 380M rising to +6.3% at 1.43B; on English ARC Easy, +9.8% at 380M rising to +19.5% at 1.43B. Notably, a practitioner who mixes in English and selects checkpoints by Arabic

VL obtains 47% on English ARC Easy at 1.43B, compared to 28% for the monolingual counterpart, without ever optimizing for English, and at no additional tuning cost (though the compute budget is shared).

HP tuning on top of mixing helps more on benchmarks than VL suggests. The value of HP tuning on top of mixing depends on the checkpoint selection rule. Under oracle selection (peak accuracy across all checkpoints), HP tuning adds little on Arabic: +1.3% at 600M and +1.4% at 1.43B. Under VL selection (accuracy at the best Arabic-VL checkpoint), the benefit grows with scale: +1.9% at 600M and +3.7% at 1.43B on Arabic; +4.9% on English at 1.43B. Optimizing directly for English under accuracy selection yields up to +7.6% at 1.43B (App. C.3). Section 4.1 showed that HP tuning on top of mixing improves Arabic VL by under 1% at every scale, so these benchmark gains are not reflected in VL. This is a first indication that Arabic VL can underestimate downstream gains, a pattern §4.4 quantifies for mixing itself.

Validation loss as a per-run checkpoint-selection proxy. The preceding comparison is *across* runs, where VL gaps understate benchmark gaps between HP configurations. Within a single run, by contrast, VL remains a reliable checkpoint selector (Fig. 3): for each bilingual run, accuracy at the best-Arabic-VL checkpoint tracks peak accuracy of that run closely, with Pearson $r = 0.97$ (Arabic) and 0.98 (English), and median absolute gaps of 0.21% and 0.13% ($n \approx 700$). Within a run, Arabic VL reliably picks near-optimal checkpoints, making it a cheap, label-free alternative to benchmark-based selection. Full statistics in Table 6 (App. C.4).

On ARC Easy, mixing is the lever that scales downstream accuracy: monolingual training plateaus near random across a $10\times$ parameter sweep, while bilingual training scales steeply on both languages. The Arabic split is machine-translated (§3), so part of the bilingual margin may reflect English-aligned artifacts rather than Arabic capability. The ordering is corroborated by Arabic validation loss (§4.1), measured on native Arabic text. HP tuning on top adds a second-order benefit partly invisible to Arabic VL, a pattern §4.4 returns to.

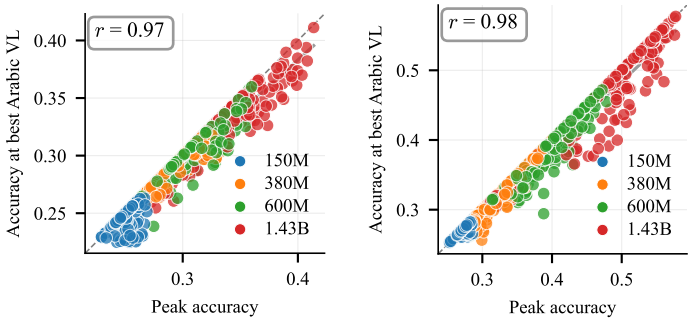


Figure 3: Accuracy at the best Arabic VL checkpoint vs. peak accuracy (5-shot). **Left**: Arabic ARC Easy ($r = 0.97$). **Right**: English ARC Easy ($r = 0.98$). Dashed line: $y = x$. Solid line: linear fit. Colors indicate model scale. Points near the diagonal confirm that Arabic validation loss selects near-optimal checkpoints for both languages.

4.3 Hyperparameter sensitivity and the case for μP

We now ask how sensitive bilingual performance is to each tuning axis, and whether the search over hyperparameters can be simplified. The optimal configuration shifts with both scale and paradigm (Table 3, App. C). For monolingual, tuned HP, the effective weight decay $\lambda_{\mu\mathrm{P}}$ climbs sharply with scale, from 0.1 at 150M to 2.0 and 2.5 at 380M and 600M before settling at 1.2 at 1.43B, compensating for overfitting on repeated Arabic data. For bilingual, tuned HP, $\lambda_{\mu\mathrm{P}}$ still grows with scale (0.1, 0.02, 0.25, 0.4 from 150M to 1.43B) but an order of magnitude more gently: *on Arabic VL, mixing in English reduces the need for aggressive weight decay*. Under tuned HPs, the optimal Arabic fraction α drops monotonically from 0.33 at 380M to 0.10 at 1.43B as the per-scale HP search finds increasingly English-heavy regimes (Fig. 9, App. C). Bilingual training with basic HP matches or beats monolingual training with tuned HP on Arabic validation loss (Table 3), confirming that the diverse signal from English is a more effective VL regularizer than weight decay alone. A practitioner adopting bilingual training therefore faces a joint search over mixing ratio and hyperparameters. We ask two questions to determine how much of that search is actually necessary.

Q1: Does the mixing ratio or the hyperparameters drive Arabic VL more? We ask whether varying $R_{\max}$ or (λ, η) produces more variation in Arabic VL across the bilingual grid, using analysis

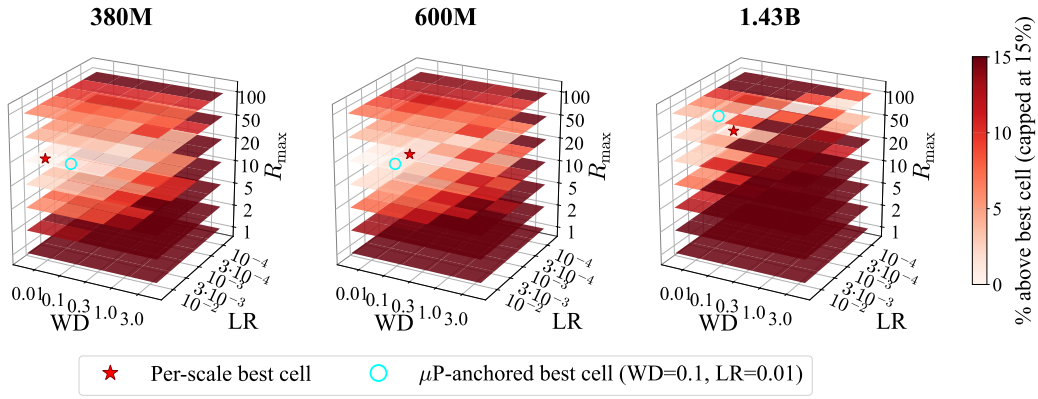


Figure 4: Arabic VL landscape over $(R_{\max}, \lambda, \eta)$ at 380M, 600M, and 1.43B, as stacked (λ, η) heatmaps along $R_{\max}$ (color: % above per-scale best, capped at 15%). Red star: per-scale best; cyan circle: μP-anchored best (λ=0.1, η=0.01). Markers share a basin at 380M/600M but separate at 1.43B.

of variance (ANOVA), which partitions the variance across a grid of configurations into per-factor contributions (App. E). The answer is not a property of the loss landscape alone, but of the landscape viewed from where the practitioner stands: it depends on which region of HP space the ANOVA covers. We consider three regions, each matching a practitioner state (Fig. 4). *(i) Top-γ% (post-hoc):* all HP cells within γ% of the best Arabic VL, wherever they sit (PED-ANOVA, Watanabe et al. (2023)); requires training the full grid (Fig. 28). *(ii) Box around the per-scale optimum (idealized):* boxes of growing radius in (λ, η) centered on the red star; requires knowing the optimum (Fig. 30, bottom). *(iii) Box around μP (a priori):* the same boxes centered on the cyan circle, the basic HPs $(\lambda$=0.1, η=0.01$)$ selected at 150M; the only view available before any per-scale training (Fig. 30, top).

View (i). At every scale and across practical thresholds, $R_{\max}$ explains most of the Arabic VL variance (Fig. 28); at 1.43B its dominance weakens only at the loosest thresholds. The full-grid view, sensitive to grid design, is in App. G. *Views (ii) and (iii).* At 380M and 600M, the red star and cyan circle share a wide low-loss basin (Fig. 4, left and middle), so both views agree: $R_{\max}$ explains >90% of the variance in the smallest box around either anchor, and the μP practitioner faces the same low HP sensitivity as one who tuned per scale. At 1.43B, the basin fragments and the anchors separate (Fig. 4, right). Around the red star, $R_{\max}$ still explains ≈89% in the smallest box; around the cyan circle, HPs and interactions account for roughly half, because the landscape near μP is sharper than at smaller scales (Fig. 5). Q2 measures whether the resulting gap to the per-scale optimum justifies per-scale tuning.

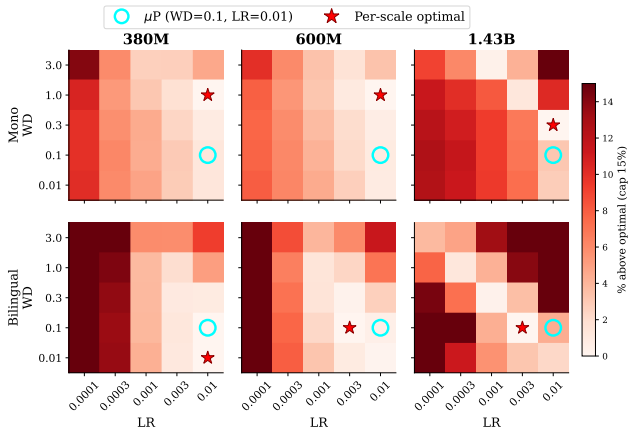


Figure 5: HP landscape (relative gap % to per-scale optimum). Top: monolingual; bottom: bilingual. Cyan circle: μP HPs; red star: per-scale optimal. At 380M/600M, uniformly pale cells indicate a flat landscape where HP choice is forgiving. At 1.43B, dark cells emerge throughout the grid, signaling sharper sensitivity. The μP cell stays within 0.6% (bilingual) and 3.1% (monolingual) of optimal at all scales. Gaps quantified in App. G.

Q2: How robust is μP transfer? Q1 recommended μP. The cost of that recommendation is minimal: transferring the best 150M proxy HPs to larger scales via μP keeps the relative gap to the per-scale tuned optimum below 0.6% on bilingual Arabic VL at every scale up to 1.43B (Fig. 37, left, App. G). The bilingual gap is 0.1% at 380M and 0.5% at both 600M and 1.43B; the monolingual gap is larger but still bounded: 1.3% at 380M, 0.9% at 600M, 3.1% at 1.43B. The heatmaps (Fig. 5) show this: the μP cell sits in a flat, near-optimal basin at 380M and 600M; at 1.43B the basin shrinks but the μP cell remains within a few percent of the per-scale optimum (App. F).

The practical recipe is therefore to transfer HPs via μP from a small proxy and spend compute on R_{max} search; a wide HP search at scale exposes the practitioner to jagged regions of HP space where a wrong choice costs substantially more (App. F).

4.4 To what extent can mixing compensate for data scarcity?

The preceding sections establish that data mixing outperforms monolingual training at $D_{\text{LR}} = 200\text{M}$ tokens, and that a practitioner using μP HPs should prioritize the mixing ratio over HP tuning. We now quantify the value of mixing: how much unique target-language data is it equivalent to? This also supplies the control §4.1 lacks: pricing mixing against monolingual models trained on more unique Arabic puts both sides on fresh data, and on validation loss mixing then loses.

Building the baseline. We train monolingual models at seven Arabic corpus sizes: $D_{\text{LR}} \in \{25\text{M}, 50\text{M}, 100\text{M}, 200\text{M}, 500\text{M}, 1\text{B}, 2\text{B}\}$ tokens, each repeated up to 100 times. For each corpus size, we take the best base (λ, η) from the 5×5 grid at 150M and transfer it to larger scales with μP (values in Table 11). Figure 6 plots the best Arabic validation loss against model size, with one curve per corpus size and the four paradigm results overlaid. On validation loss, more unique data dominates: 2.5× more Arabic data alone already matches the best bilingual result for the 600M model.

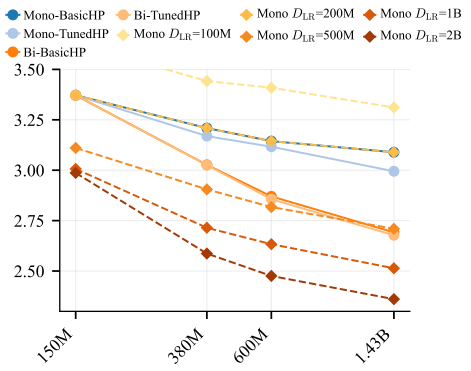


Figure 6: Best Arabic validation loss vs. model size. Dashed lines (yellow to brown): monolingual training at each Arabic corpus size D_{LR} (100M–2B tokens). Solid lines: the four paradigm results at $D_{\text{LR}} = 200\text{M}$ (blue: monolingual, orange: bilingual; dark: basic HP, light: tuned HP). On VL, more unique Arabic data consistently beats mixing: the mono D_{LR}=500M curve already sits below bilingual, tuned HP at 600M and above.

The data multiplier. To quantify the value of mixing, we define the *data multiplier* $N = D_{\text{LR}}^{\text{equiv}}/200\text{M}$. The denominator is the Arabic budget of the configuration being evaluated (e.g., bilingual with tuned HPs, trained on 200M Arabic tokens plus English); $D_{\text{LR}}^{\text{equiv}}$ is the unique Arabic corpus a monolingual μP model would need to match it. We compute N by fitting a log-linear model to the monolingual baseline sweep and inverting it at the bilingual target (App. D). We report N under two conditions. In the *controlled* comparison, both sides follow the same rule: each model uses the base (λ, η) with the best Arabic VL at 150M on its own training data, transferred with μP. This yields $(\lambda{=}0.1, \eta{=}0.01)$ for the bilingual model and for the monolingual reference at $D_{\text{LR}} = 200\text{M}$; the other monolingual references use the values selected for their corpus size (Table 11). In the *optimized* comparison, the bilingual model additionally gets a full HP sweep over $\lambda \times \eta \times R_{\text{max}}$.

On ARC Easy, the multiplier diverges between validation loss and benchmark accuracy. On VL, the multiplier is modest: 2.1× at 380M, 2.6× at 600M, 3.2× at 1.43B under controlled HPs (Fig. 7, left). On ARC Easy, it grows steeply with model size: 2.0× at 380M, 3.1× at 600M, 8.1× (1.6B Arabic tokens) at 1.43B (Fig. 7, right). HP tuning on top raises this to 13.4× (2.7B Arabic tokens), whereas monolingual HP tuning is worth only ∼1.5×. On SciQ and PIQA, the ordering holds at 1.43B: bilingual μP reaches 3–15×, monolingual tuning 1.3–6.3× (App. D.4). The divergence from VL holds on PIQA and open-book SciQ but not on closed-book SciQ, which stays at 1.3–4.7×, close

to the VL multiplier; only PIQA also replicates the growth with model size ($2.0 \rightarrow 15.0\times$ at 5-shot). Where VL saturates but the benchmark has headroom, VL credits only part of mixing's gain.

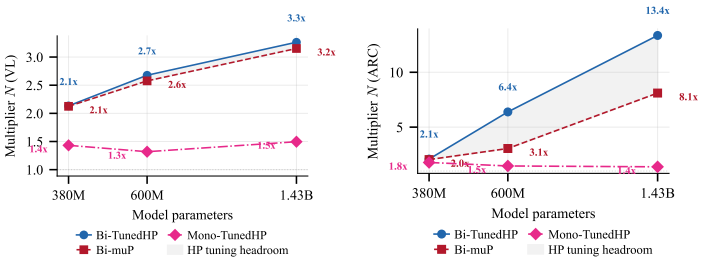


Figure 7: Data multiplier N vs. model size. **Left**: Arabic VL. **Right**: ARC Easy Arabic (5-shot). On VL, N stays within 1.3–3.3×; on the benchmark, bilingual reaches 13.4× while monolingual stays below 2×. VL-based checkpoint selection: App. D.3.

The benchmark flip. On validation loss, more unique Arabic data always beats mixing (Fig. 6). On benchmarks, the ranking flips (Fig. 8). Repeating Arabic does not close the gap: for the 1.43B model, monolingual training on the same 200M tokens sees 18.4B Arabic tokens and reaches 33.6% accuracy; bilingual training sees only 9.2B Arabic tokens and reaches 41.4%. More unique Arabic nearly does: monolingual training on 10× more unique Arabic ($D_{\text{LR}} = 2\text{B}$) is 0.6 points ahead. For English, bilingual leads at every D_{LR}: 57.7% vs. 29.6% at 200M.

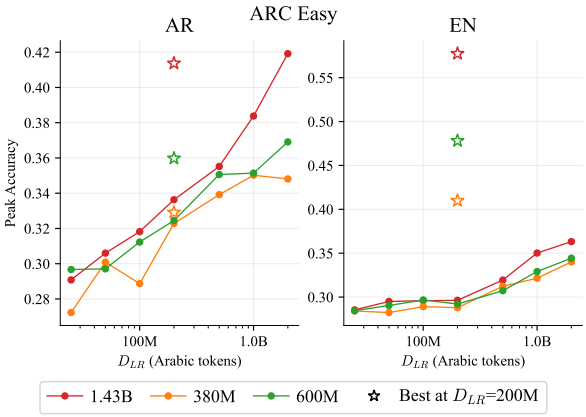


Figure 8: Peak ARC Easy accuracy (5-shot) vs. Arabic data budget (D_{LR}). **Lines**: monolingual training at each model scale. **Stars**: best bilingual configuration at $D_{\text{LR}} = 200\text{M}$ with per-scale tuned HPs. **Left** (Arabic): bilingual stars approach the high-D_{LR} end of each monolingual curve; for the 1.43B model, bilingual comes within 0.6 points of monolingual at $D_{\text{LR}} = 2\text{B}$, which uses 10× more unique Arabic. **Right** (English): bilingual dominates at every D_{LR}. μP variant: Fig. 18.

A plausible explanation is that the two metrics reward different things. Repetition lets the model overfit to in-distribution patterns, which lowers VL on a held-out sample of the same distribution without improving general capabilities; benchmarks require knowledge and skills largely absent from 200M Arabic tokens but abundant in English. English data therefore plays two roles (Seto et al., 2025): it *regularizes* by diversifying the training signal, and it *adds knowledge and skills* the Arabic corpus lacks. Part of this is reading comprehension rather than stored knowledge: mixing gains most on SciQ when the supporting passage is given in context, and least when the question is asked alone (App. D.4). VL appears to capture only the first; benchmarks capture both. Practically, VL is a reasonable proxy *within* a paradigm but should not be used alone to rank monolingual against bilingual models.

5 Conclusion

We studied how data mixing, hyperparameter tuning, and data scaling compare when pre-training language models for a data-constrained target language, using Arabic subsampled to 200M unique tokens as a controlled stand-in and English as the auxiliary. Three findings emerge. First, mixing improves Arabic validation loss more than hyperparameter tuning for models of 380M parameters and above, and the gap grows with model size, reaching 12.9% against 3.1% for the 1.43B-parameter model. Tuning on top of mixing adds under 1%, and μP transfer from a 150M-parameter proxy stays within 0.6% of per-scale tuning. Second, mixing is worth the equivalent of 2–3× the unique Arabic data on validation loss and 2–13× on ARC Easy accuracy; for the 1.43B-parameter model, this is up to 2.7B unique Arabic tokens. Third, validation loss can undercount mixing's value: it credits the regularization effect but not the knowledge the auxiliary language brings in.

AI USE STATEMENT

We used generative AI tools to implement methods (training, evaluation, analysis, and plotting code), to brainstorm and give feedback on the experimental design, and to help interpret results. The Arabic splits of ARC Easy, SciQ, and PIQA were machine-translated with NLLB-200-3.3B (§3). Formulating mathematical claims and writing proofs are not applicable to this work. Additionally, we used generative AI tools to draft and edit parts of the paper. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language, 2019. URL https://arxiv.org/abs/1911.11641.

Tyler A. Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K. Bergen. When is multilinguality a curse? language modeling for 250 high- and low-resource languages, 2023. URL https://arxiv.org/abs/2311.09205.

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, 2018. URL https://arxiv.org/abs/1803.05457.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale, 2020. URL https://arxiv.org/abs/1911.02116.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. How abilities in large language models are affected by supervised fine-tuning data composition, 2024. URL https://arxiv.org/abs/2310.05492.

Samir Yitzhak Gadre, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman, Rulin Shao, Jean Mercat, Alex Fang, Jeffrey Li, Sedrick Keh, Rui Xin, Marianna Nezhurina, Igor Vasiljevic, Jenia Jitsev, Luca Soldaini, Alexandros G. Dimakis, Gabriel Ilharco, Pang Wei Koh, Shuran Song, Thomas Kollar, Yair Carmon, Achal Dave, Reinhard Heckel, Niklas Muennighoff, and Ludwig Schmidt. Language models scale reliably with over-training and on downstream tasks, 2024. URL https://arxiv.org/abs/2403.08540.

Yifei He, Alon Benhaim, Barun Patra, Praneetha Vaddamanu, Sanchit Ahuja, Parul Chopra, Vishrav Chaudhary, Han Zhao, and Xia Song. Scaling laws for multilingual language models, 2024. URL https://arxiv.org/abs/2410.12883.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, Scott Johnston, Ben Mann, Chris Olah, Catherine Olsson, Dario Amodei, Nicholas Joseph, Jared Kaplan, and Sam McCandlish. Scaling laws and interpretability of learning from repeated data, 2022. URL https://arxiv.org/abs/2205.10487.

Ferenc Huszár. How (not) to train your generative model: Scheduled sampling, likelihood, adversary?, 2015. URL https://arxiv.org/abs/1511.05101.

Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. An efficient approach for assessing hyperparameter importance. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 754–762, Bejing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/hutter14.html.

Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations, 2024. URL https://arxiv.org/abs/2405.18392.

Konwoo Kim, Suhas Kotha, Percy Liang, and Tatsunori Hashimoto. Pre-training under infinite compute, 2025. URL https://arxiv.org/abs/2509.14786.

Xize Liang, Lin Yang, Jie Wang, Yiyang Lu, Runyu Wu, Hanzhu Chen, and Jianye HAO. Boosting multi-domain fine-tuning of large language models through evolving interactions between samples. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Si0HHbjBfU.

Peiqin Lin, Shaoxiong Ji, Jörg Tiedemann, André F. T. Martins, and Hinrich Schütze. Mala-500: Massive language adaptation of large language models, 2024. URL https://arxiv.org/abs/2401.13303.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. Few-shot learning with multilingual language models, 2022. URL https://arxiv.org/abs/2112.10668.

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, 2025. URL https://arxiv.org/abs/2407.01492.

Shayne Longpre, Sneha Kudugunta, Niklas Muennighoff, I-Hung Hsu, Isaac Caswell, Alex Pentland, Sercan Arik, Chen-Yu Lee, and Sayna Ebrahimi. Atlas: Adaptive transfer scaling laws for multilingual pretraining, finetuning, and decoding the curse of multilinguality, 2026. URL https://arxiv.org/abs/2510.22037.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models, 2025. URL https://arxiv.org/abs/2305.16264.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know?, 2019. URL https://arxiv.org/abs/1810.09136.

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, 2024. URL https://arxiv.org/abs/2406.17557.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. Fineweb2: One pipeline to scale them all – adapting pre-training data processing to every language, 2025. URL https://arxiv.org/abs/2506.20920.

Tomer Porian, Mitchell Wortsman, Jenia Jitsev, Ludwig Schmidt, and Yair Carmon. Resolving discrepancies in compute-optimal scaling of language models, 2025. URL https://arxiv.org/abs/2406.19146.

Philipp Probst, Bernd Bischl, and Anne-Laure Boulesteix. Tunability: Importance of hyperparameters of machine learning algorithms, 2018. URL https://arxiv.org/abs/1802.09596.

Anastasiia Sedova, Skyler Seto, Natalie Schluter, and Pierre Ablin. Scaling laws for mixture pretraining under data constraints, 2026. URL https://arxiv.org/abs/2605.12715.

Skyler Seto, Maartje ter Hoeve, Richard He Bai, Natalie Schluter, and David Grangier. Training bilingual lms with data constraints in the targeted language, 2025. URL https://arxiv.org/abs/2411.12986.

Mustafa Shukor, Louis Bethune, Dan Busbridge, David Grangier, Enrico Fini, Alaaeldin El-Nouby, and Pierre Ablin. Scaling laws for optimal data mixtures, 2025. URL https://arxiv.org/abs/2507.09404.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. No language left behind: Scaling human-centered machine translation, 2022. URL https://arxiv.org/abs/2207.04672.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models, 2016. URL https://arxiv.org/abs/1511.01844.

Shuhei Watanabe, Archit Bansal, and Frank Hutter. Ped-anova: Efficiently quantifying hyperparameter importance in arbitrary subspaces, 2023. URL https://arxiv.org/abs/2304.10255.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions, 2017. URL https://arxiv.org/abs/1707.06209.

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, 2023. URL https://arxiv.org/abs/2305.10429.

Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer, 2022. URL https://arxiv.org/abs/2203.03466.

Kazuki Yano, Shun Kiyono, Sosuke Kobayashi, Sho Takase, and Jun Suzuki. Pre-training llm without learning rate decay enhances supervised fine-tuning, 2026. URL https://arxiv.org/abs/2603.16127.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance, 2025. URL https://arxiv.org/abs/2403.16952.

A Learning rate schedule

All experiments use a constant learning rate with no warmup and no decay. Constant or near-constant schedules are established practice in recent pre-training work: Hägele et al. (2024) show that constant LR with a short cooldown matches cosine at matched compute; Porian et al. (2025) reproduce the Chinchilla scaling law under a constant schedule, finding that learning-rate decay is not essential for its validity; and Yano et al. (2026) pre-train 1B and 8B models under a constant LR through the end of training.

Our design also requires it. The central object of analysis is a loss-vs-repetition curve read along a single training run (Figures 1, 6). A decay schedule couples the learning rate to the training horizon, so checkpoints at different repetition counts R sit at different points on the schedule and are not directly comparable within a run. Matching our sweep under a decay schedule would require one run per repetition point, multiplying compute by the length of the R axis.

The cost is that absolute loss would be lower under a schedule with a cooldown. All four paradigms share the same constant schedule, so cross-paradigm comparisons and the relative conclusions reported in the paper are unaffected.

B Model architectures and training configurations

All models are LLaMA-style decoder-only transformers with SiLU activations, RMSNorm ($\epsilon = 10^{-5}$), rotary position embeddings (RoPE, base 10,000), and tied input/output embeddings. We use the XGLM tokenizer (Lin et al., 2022) with a 256K vocabulary. Table 2 summarizes the architecture at each scale.

Table 2: Model architectures. All models use a head dimension of 128 and a context length of 2048 tokens.

	150M	380M	600M	1.43B
Layers	6	8	12	16
Hidden dim	512	1024	1280	2048
Attn heads	4	8	10	16
KV heads	4	8	10	8
FFN dim	1792	3584	4480	7168
Parameters	154M	384M	613M	1.43B

Optimizer. All models use AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$, gradient clipping at norm 1.0, and bfloat16 precision. We use a constant learning rate schedule. An auxiliary z-loss with weight 10^{-4} stabilizes training.

μP scaling. We use maximal update parameterization with a base dimension of 512 (the 150M model). The scaling factor $m_N = d_{\text{model}}/512$ adjusts the learning rate ($\eta' = \eta_{\text{base}}/m_N$) and weight decay ($\lambda' = \lambda_{\text{base}} \times m_N$) for hidden-layer parameters. Base hyperparameters are $\eta = 10^{-2}$ and $\lambda = 0.1$, tuned at the 150M proxy scale and transferred to larger models.

C Per-paradigm detailed results

C.1 Optimal hyperparameters and Arabic fraction

Table 3 reports the optimal hyperparameters for each paradigm and model scale, selected by best Arabic validation loss. All values are μP base hyperparameters (before scaling by m_N).

Table 3 reports the configurations that minimize Arabic validation loss. Figure 9 visualizes the optimal Arabic fraction α^* from this table as a function of model size, showing a power-law decay in both HP regimes.

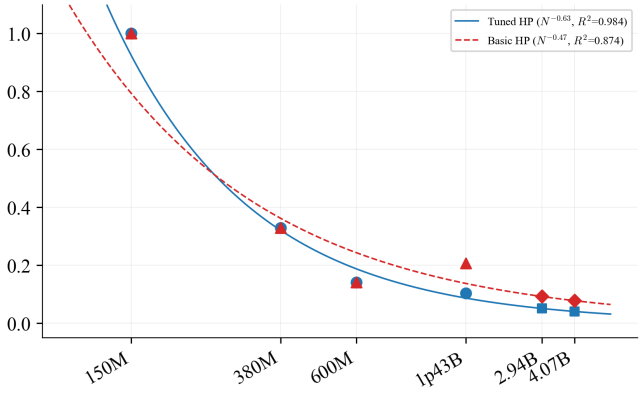


Figure 9: Optimal Arabic fraction α^* vs. model size. Both HP regimes follow a power-law decay, with the tuned-HP curve declining faster ($\beta = 0.63$) than basic HP ($\beta = 0.47$). Extrapolations to 2.94B and 4.07B shown.

Table 3: Optimal hyperparameters per paradigm and model scale (Arabic validation loss). λ and η are μP base values; $\lambda_{\mu\text{P}}$ and $\eta_{\mu\text{P}}$ are the effective values after μP scaling ($\eta_{\mu\text{P}} = \eta/m_N$, $\lambda_{\mu\text{P}} = \lambda \times m_N$, where $m_N = d_{\text{model}}/512$). R^*: repetition count at the best checkpoint. R_{max}: maximum repetition budget (bilingual only). α: Arabic fraction at the selected R_{max} (bilingual only). All 150M bilingual runs select $\alpha = 1$ (monolingual), confirming that mixing does not help at this scale.

Paradigm	Scale	λ	$\lambda_{\mu\text{P}}$	η	$\eta_{\mu\text{P}}$	R^*	R_{max}	α	Loss
Monolingual, basic	150M	0.1	0.1	0.01	0.01	69	—	—	3.372
	380M	0.1	0.2	0.01	0.005	27	—	—	3.209
	600M	0.1	0.25	0.01	0.004	21	—	—	3.144
	1.43B	0.1	0.4	0.01	0.0025	7	—	—	3.089
Monolingual, tuned	150M	0.1	0.1	0.01	0.01	69	—	—	3.372
	380M	1.0	2.0	0.01	0.005	47	—	—	3.169
	600M	1.0	2.5	0.01	0.004	39	—	—	3.116
	1.43B	0.3	1.2	0.01	0.0025	23	—	—	2.995
Bilingual, basic	150M	0.1	0.1	0.01	0.01	69	100	1.00	3.372
	380M	0.1	0.2	0.01	0.005	20	20	0.33	3.027
	600M	0.1	0.25	0.01	0.004	20	20	0.14	2.870
	1.43B	0.1	0.4	0.01	0.0025	77	100	0.21	2.692
Bilingual, tuned	150M	0.1	0.1	0.01	0.01	69	100	1.00	3.372
	380M	0.01	0.02	0.01	0.005	20	20	0.33	3.025
	600M	0.1	0.25	0.003	0.0012	20	20	0.14	2.856
	1.43B	0.1	0.4	0.003	0.00075	50	50	0.10	2.678

C.2 Optimizing for English instead of Arabic

So far every result selects hyperparameters and checkpoints to minimize Arabic validation loss. We now flip the objective. Figure 10 reports the best English validation loss achievable by each paradigm when HPs and checkpoints are selected to minimize English VL. Figure 11 reports the Arabic validation loss attained at those same English-optimal configurations, quantifying the cost in Arabic performance when a practitioner prioritizes the auxiliary language.

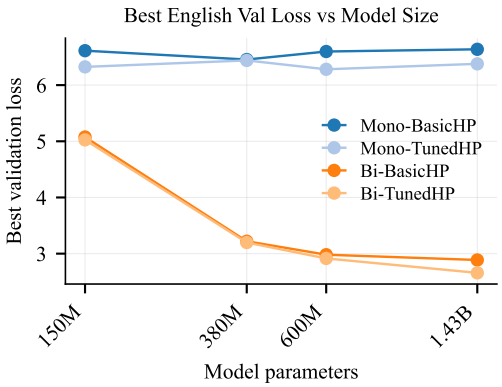


Figure 10: Best English validation loss vs. model size for each paradigm, selecting the checkpoint and HPs that minimize English VL. Monolingual Arabic models (top two curves) show essentially flat English loss across scales (6.3–6.6), since they never see English tokens. Bilingual models improve English dramatically with scale, from ∼5.1 at 150M to 2.66 at 1.43B with tuned HPs. HP tuning provides a modest additional gain in the bilingual setting (2.89 → 2.66 at 1.43B).

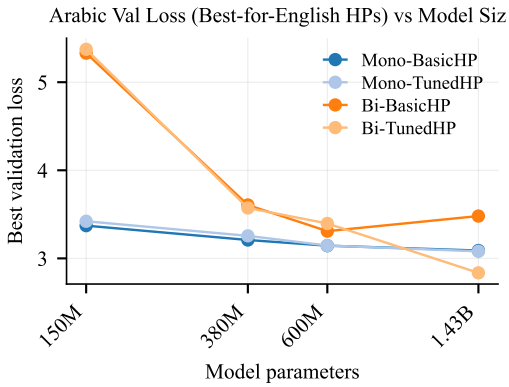


Figure 11: Arabic validation loss at the best-English checkpoint vs. model size. For monolingual runs, the best-English checkpoint corresponds to fewer repetitions than the best-Arabic one (e.g. $R = 10$ vs. $R = 27$ at 380M for Monolingual, basic), so Arabic loss is slightly worse. For bilingual runs, the best-English HPs select $R_{\max} \leq 2$ at all scales below 1.43B ($\alpha < 0.09$), producing high Arabic loss at small scales. At 1.43B with tuned HPs, the gap narrows: best-English bilingual achieves Arabic loss of 2.84 vs. 2.68 when optimizing directly for Arabic, confirming that Arabic and English objectives converge at larger scale with proper tuning.

C.3 Benchmark accuracy by paradigm

This subsection breaks down the summary in Fig. 2 by paradigm, scale, language, and checkpoint-selection rule. Tables 4 (Arabic) and 5 (English) report ARC Easy 5-shot accuracy under two rules: best accuracy across all checkpoints (left columns, oracle) and accuracy at the checkpoint minimizing Arabic VL (right columns, the cheap practitioner rule). Figures 12 and 13 plot the same numbers against model size. Three patterns recur across every panel: (i) at 150M all four paradigms collapse to the same accuracy, since the model consumes the full 200M Arabic budget without overfitting; (ii) monolingual accuracy is near-flat with scale on both languages while bilingual scales steeply; (iii) HP tuning adds a small, consistent margin on top of mixing, larger under oracle selection than under VL selection.

Table 4: ARC Easy Arabic (5-shot) accuracy per paradigm and model scale. **Left columns**: best accuracy across all checkpoints (selected by accuracy). **Right columns**: accuracy at the checkpoint with best Arabic validation loss (selected by VL).

	Selected by accuracy				Selected by Arabic VL			
Paradigm	150M	380M	600M	1.43B	150M	380M	600M	1.43B
Monolingual, basic HP	.311	.323	.325	.336	.297	.304	.306	.311
Monolingual, tuned HP	.311	.326	.334	.351	.297	.302	.305	.323
Bilingual, basic HP	.311	.329	.347	.400	.297	.308	.327	.374
Bilingual, tuned HP	.311	.329	.360	.414	.297	.318	.346	.411

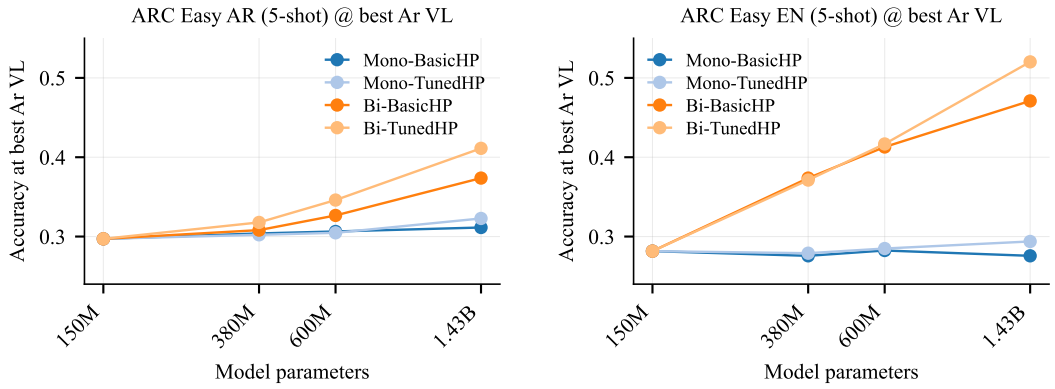


Figure 12: ARC Easy accuracy (5-shot) vs. model size at the checkpoint with best Arabic validation loss (dashed lines in Fig. 2). **Left**: Arabic. **Right**: English. Blue: monolingual; orange: bilingual (dark = basic HP, light = tuned HP). Monolingual training is flat; bilingual scales steeply on both languages.

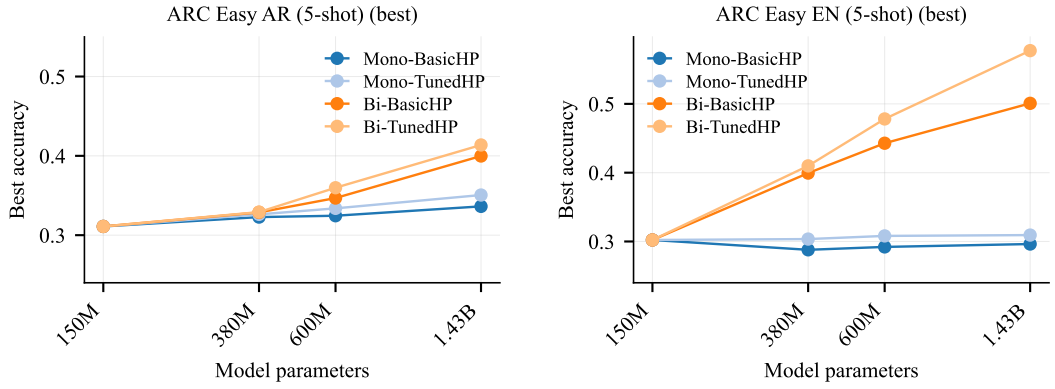


Figure 13: ARC Easy accuracy (5-shot) vs. model size under oracle selection (best accuracy across all checkpoints; solid lines in Fig. 2). **Left**: Arabic. **Right**: English. HP tuning provides a larger benefit under oracle selection, especially on English at 1.43B (+7.7%), but this requires optimizing for each language separately.

Table 5: ARC Easy English (5-shot) accuracy per paradigm and model scale. Same layout as Table 4. Monolingual English accuracy is flat and near random (∼28–30%). Bilingual training scales steeply.

	Selected by accuracy				Selected by Arabic VL			
Paradigm	150M	380M	600M	1.43B	150M	380M	600M	1.43B
Monolingual, basic HP	.302	.288	.292	.296	.282	.276	.282	.276
Monolingual, tuned HP	.302	.304	.308	.309	.282	.279	.285	.294
Bilingual, basic HP	.302	.399	.443	.501	.282	.374	.413	.471
Bilingual, tuned HP	.302	.410	.478	.577	.282	.371	.417	.520

C.4 Validation loss as a per-run checkpoint selector

Validation loss as a proxy. VL is a better model selection proxy for bilingual models than for monolingual ones. In monolingual training, the mean VL-accuracy gap is 2.1% for Arabic and 1.8% for English, with peak accuracy occurring after the best-VL checkpoint 84% of the time. In bilingual training, the mean gap drops to 0.6%, and peak accuracy occurs before or at the best-VL checkpoint 87% of the time.

Validation loss proxy: full statistics. Table 6 reports Pearson r, RMSE, and median absolute gap for all four evaluation settings, broken down by model scale. Figure 14 shows the 0-shot version of the scatter plots (5-shot is in the main text, Fig. 3).

Table 6: Validation loss as a model selection proxy: Pearson r, RMSE (%), and median |gap| (%) between accuracy at the best Arabic VL checkpoint and peak accuracy, per evaluation setting and model scale.

Setting	Metric	150M	380M	600M	1.43B
AR 0-shot ($r = 0.98$)	RMSE (%)	0.84	0.64	1.04	1.13
	Median |gap| (%)	0.19	0.00	0.04	0.30
AR 5-shot ($r = 0.97$)	RMSE (%)	1.16	0.88	1.27	1.53
	Median |gap| (%)	0.19	0.00	0.42	0.42
EN 0-shot ($r = 0.99$)	RMSE (%)	0.60	0.77	1.50	1.84
	Median |gap| (%)	0.13	0.00	0.13	0.38
EN 5-shot ($r = 0.98$)	RMSE (%)	0.68	1.05	1.94	2.77
	Median |gap| (%)	0.19	0.00	0.13	0.42

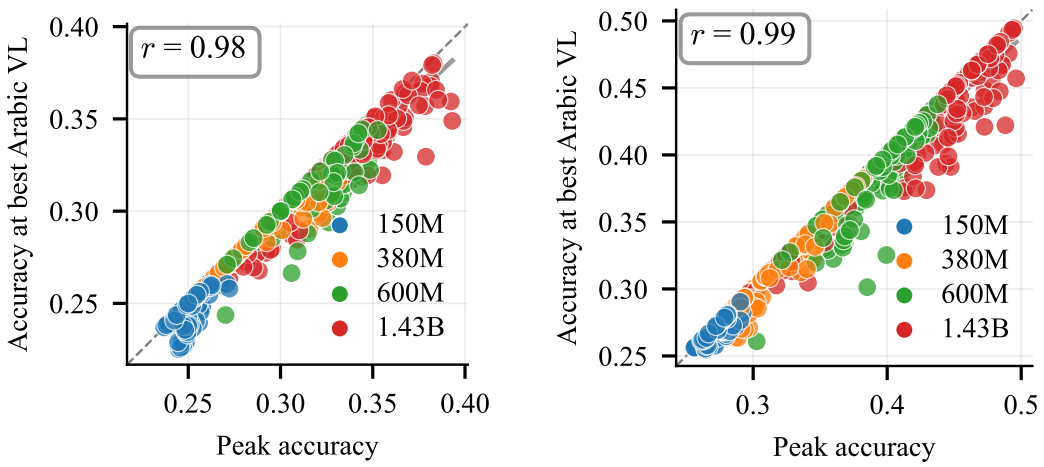


Figure 14: Accuracy at the best Arabic VL checkpoint vs. peak accuracy (0-shot). **Left**: Arabic ARC Easy ($r = 0.98$). **Right**: English ARC Easy ($r = 0.99$). Same format as Fig. 3.

C.5 Labeled-point configurations

Tables 7–10 list the configuration (λ, η, $R_{\max}$) and accuracy for each labeled point in Figures 12 and 13. λ and η are base (pre-μP) values.

Table 7: Arabic ARC Easy points, selected by best accuracy (Fig. 13, left).

ID	Paradigm	Model	λ	η	R	Acc
1	Mono, basic	150M	0.1	0.01	100	0.311
2	Mono, basic	380M	0.1	0.01	100	0.323
3	Mono, basic	600M	0.1	0.01	100	0.325
4	Mono, basic	1.43B	0.1	0.01	100	0.336
5	Mono, tuned	150M	0.1	0.01	100	0.311
6	Mono, tuned	380M	0.01	0.003	100	0.326
7	Mono, tuned	600M	3	0.001	100	0.334
8	Mono, tuned	1.43B	1	0.003	100	0.351
9	Bi, basic	150M	0.1	0.01	100	0.311
10	Bi, basic	380M	0.1	0.01	50	0.329
11	Bi, basic	600M	0.1	0.01	20	0.347
12	Bi, basic	1.43B	0.1	0.01	100	0.400
13	Bi, tuned	150M	0.1	0.01	100	0.311
14	Bi, tuned	380M	1	0.003	50	0.329
15	Bi, tuned	600M	0.3	0.001	20	0.360
16	Bi, tuned	1.43B	0.1	0.003	50	0.414

Table 8: Arabic ARC Easy points, selected by best Arabic VL (Fig. 12, left).

ID	Paradigm	Model	λ	η	R	Acc
1	Mono, basic	150M	0.1	0.01	100	0.297
2	Mono, basic	380M	0.1	0.01	100	0.304
3	Mono, basic	600M	0.1	0.01	100	0.306
4	Mono, basic	1.43B	0.1	0.01	100	0.311
5	Mono, tuned	150M	0.1	0.01	100	0.297
6	Mono, tuned	380M	1	0.01	100	0.302
7	Mono, tuned	600M	1	0.01	100	0.305
8	Mono, tuned	1.43B	0.3	0.01	100	0.323
9	Bi, basic	150M	0.1	0.01	100	0.297
10	Bi, basic	380M	0.1	0.01	20	0.308
11	Bi, basic	600M	0.1	0.01	20	0.327
12	Bi, basic	1.43B	0.1	0.01	100	0.374
13	Bi, tuned	150M	0.1	0.01	100	0.297
14	Bi, tuned	380M	0.01	0.01	20	0.318
15	Bi, tuned	600M	0.1	0.003	20	0.346
16	Bi, tuned	1.43B	0.1	0.003	50	0.411

Table 9: English ARC Easy points, selected by best accuracy (Fig. 13, right).

ID	Paradigm	Model	λ	η	R	Acc
1	Mono, basic	150M	0.1	0.01	100	0.302
2	Mono, basic	380M	0.1	0.01	100	0.288
3	Mono, basic	600M	0.1	0.01	100	0.292
4	Mono, basic	1.43B	0.1	0.01	100	0.296
5	Mono, tuned	150M	0.1	0.01	100	0.302
6	Mono, tuned	380M	1	0.003	100	0.304
7	Mono, tuned	600M	1	0.003	100	0.308
8	Mono, tuned	1.43B	3	0.001	100	0.309
9	Bi, basic	150M	0.1	0.01	100	0.302
10	Bi, basic	380M	0.1	0.01	1	0.399
11	Bi, basic	600M	0.1	0.01	10	0.443
12	Bi, basic	1.43B	0.1	0.01	50	0.501
13	Bi, tuned	150M	0.1	0.01	100	0.302
14	Bi, tuned	380M	0.01	0.003	1	0.410
15	Bi, tuned	600M	0.01	0.01	5	0.478
16	Bi, tuned	1.43B	0.01	0.001	1	0.577

Table 10: English ARC Easy points, selected by best Arabic VL (Fig. 12, right). Same checkpoints as Table 8.

ID	Paradigm	Model	λ	η	R	Acc
1	Mono, basic	150M	0.1	0.01	100	0.282
2	Mono, basic	380M	0.1	0.01	100	0.276
3	Mono, basic	600M	0.1	0.01	100	0.282
4	Mono, basic	1.43B	0.1	0.01	100	0.276
5	Mono, tuned	150M	0.1	0.01	100	0.282
6	Mono, tuned	380M	1	0.01	100	0.279
7	Mono, tuned	600M	1	0.01	100	0.285
8	Mono, tuned	1.43B	0.3	0.01	100	0.294
9	Bi, basic	150M	0.1	0.01	100	0.282
10	Bi, basic	380M	0.1	0.01	20	0.374
11	Bi, basic	600M	0.1	0.01	20	0.413
12	Bi, basic	1.43B	0.1	0.01	100	0.471
13	Bi, tuned	150M	0.1	0.01	100	0.282
14	Bi, tuned	380M	0.01	0.01	20	0.371
15	Bi, tuned	600M	0.1	0.003	20	0.417
16	Bi, tuned	1.43B	0.1	0.003	50	0.520

D Data multiplier and equivalence

D.1 Monolingual training across Arabic corpus sizes

We train monolingual models at seven Arabic corpus sizes, $D_{\text{LR}} \in \{25\text{M}, 50\text{M}, 100\text{M}, 200\text{M}, 500\text{M}, 1\text{B}, 2\text{B}\}$ tokens, each repeated up to 100 times, with μP HPs. We report Arabic validation loss (at 150M, 380M, 600M, 1.43B) and ARC Easy 5-shot accuracy (at 380M, 600M, 1.43B).

Validation loss. Table 11 reports the best Arabic VL at each (D_{LR}, scale) pair alongside the optimal repetition count R^* at which that minimum is reached. Figure 16 plots the same loss against D_{LR}. VL decreases steadily with more unique data at every scale, with gains diminishing in the high-D_{LR} regime.

Table 11: Best Arabic validation loss for monolingual training at each corpus size. HPs are the best base values from the 5×5 grid at 150M for that corpus size, transferred with μP: $\eta = 0.01$ throughout, and $\lambda = 0.3, 1.0, 0.3, 0.1, 0.01, 0.01, 0.01$ from 25M to 2B. R^* is the optimal repetition count.

	150M		380M		600M		1.43B	
D_{LR}	Loss	R^*	Loss	R^*	Loss	R^*	Loss	R^*
25M	4.698	100	4.177	79	4.140	55	4.262	24
50M	4.060	100	3.817	59	3.661	100	3.593	35
100M	3.629	95	3.442	34	3.409	28	3.311	12
200M	3.372	69	3.209	27	3.144	21	3.089	7
500M	3.111	60	2.905	20	2.817	14	2.709	8
1B	3.007	31	2.715	25	2.633	15	2.514	6
2B	2.987	15	2.587	47	2.476	25	2.361	8

Optimal repetition count. Figure 15 visualizes how R^* shrinks as the corpus grows: the 1.43B model peaks at $R^* = 24$ with 25M tokens but at $R^* = 8$ with 2B tokens. More unique data means less repetition is needed before overfitting dominates.

Downstream accuracy. Table 12 reports peak ARC Easy 5-shot accuracy for the same sweep, split by language. Arabic accuracy rises from 27–30% at $D_{\text{LR}} = 25$M to 35–41% at $D_{\text{LR}} = 2$B. English accuracy also rises modestly with corpus size even though no dedicated English data is seen. These monolingual numbers are the reference against which bilingual accuracy at $D_{\text{LR}} = 200$M is compared in the multiplier analysis: how much monolingual Arabic data is equivalent to the 200M-token bilingual run?

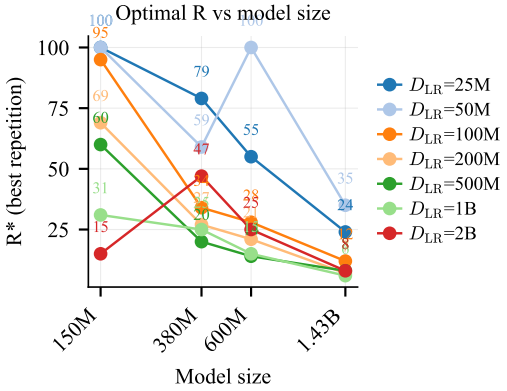


Figure 15: Optimal repetition count R^* vs. Arabic corpus size D_{LR} for each model scale.

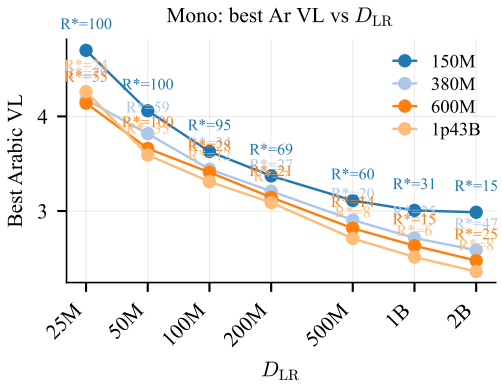


Figure 16: Best Arabic validation loss vs. Arabic corpus size D_{LR} for each model scale (monolingual, HPs as in Table 11).

Table 12: ARC Easy accuracy (5-shot) for monolingual training at each Arabic corpus size and model scale.

	Arabic			English		
D_{LR}	380M	600M	1.43B	380M	600M	1.43B
25M	0.272	0.297	0.291	0.284	0.285	0.285
50M	0.301	0.297	0.306	0.282	0.290	0.295
100M	0.289	0.312	0.318	0.289	0.297	0.296
200M	0.323	0.325	0.336	0.288	0.292	0.296
500M	0.339	0.351	0.355	0.312	0.307	0.319
1B	0.350	0.351	0.384	0.322	0.329	0.350
2B	0.348	0.368	0.410	0.340	0.344	0.363

D.2 Variants: 0-shot and μP peak-accuracy plots

Figure 8 in the main body plots peak ARC Easy 5-shot accuracy against D_{LR}, with per-scale monolingual curves and bilingual-tuned-HP stars at $D_{\text{LR}} = 200$M sitting above those curves. That layout is the visual hint behind the multiplier: the horizontal gap between a star and the corresponding monolingual curve is $D_{\text{LR}}^{\text{equiv}}$, and the multiplier simply reports it as a ratio to 200M. The three panels below repeat the same layout under two natural variants, 0-shot evaluation and μP (rather than per-scale-tuned) HPs for the bilingual stars.

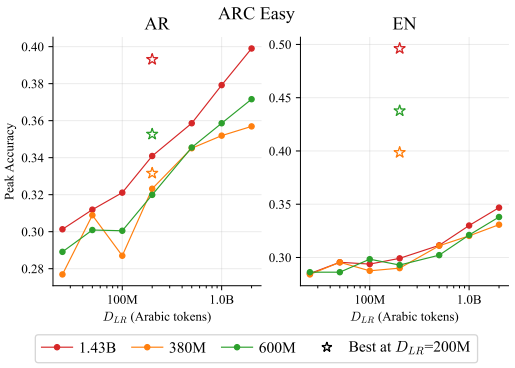


Figure 17: Peak ARC Easy accuracy (0-shot) vs. Arabic data budget (D_{LR}). Same layout as Fig. 8 but with 0-shot evaluation; stars use per-scale tuned HPs. The Arabic-side pattern mirrors 5-shot: bilingual approaches but does not exceed monolingual at $D_{\text{LR}} = 2$B; English-side bilingual dominance persists.

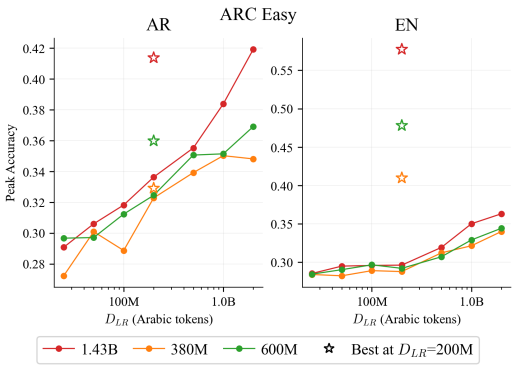


Figure 18: Peak ARC Easy accuracy (5-shot) vs. Arabic data budget (D_{LR}) with μP HPs for the bilingual runs. Same layout and monolingual lines as Fig. 8; only the bilingual stars change. Compared with tuned HPs (Fig. 8), μP stars sit slightly below on Arabic and on English at 1.43B, confirming that per-scale tuning gains a few accuracy points on top of μP transfer.

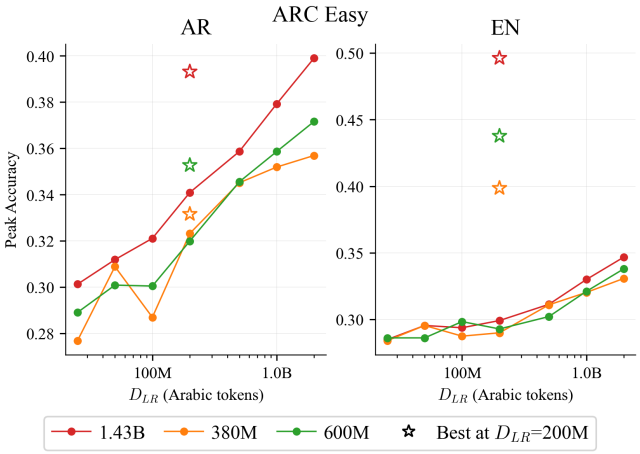


Figure 19: Peak ARC Easy accuracy (0-shot) vs. Arabic data budget (D_{LR}) with μP HPs for the bilingual runs. Same layout as Fig. 18 but with 0-shot evaluation.

D.3 Data multiplier construction and results

Multiplier construction. Recall from §4.4 that the multiplier is $N = D_{\text{LR}}^{\text{equiv}}/200\text{M}$: the unique Arabic corpus size a monolingual model would need to match a bilingual model trained on only 200M Arabic tokens. To compute it on a given metric (Arabic validation loss or ARC Easy accuracy), we need two ingredients: (i) a reference curve mapping D_{LR} to best monolingual performance, and (ii) a bilingual target value to invert against. The reference curve is built from the monolingual sweep in §D.1: for each $D_{\text{LR}} \in \{25\text{M}, 50\text{M}, 100\text{M}, 200\text{M}, 500\text{M}, 1\text{B}, 2\text{B}\}$ we take the best performance achieved over repetitions, using the per-corpus-size HPs of Table 11, then fit a log-linear model $y = a\ln(D_{\text{LR}}) + b$ to the seven points. Fits are tight at every scale ($r^2 \geq 0.90$; coefficients in Table 13). The bilingual target is the best value achieved at $D_{\text{LR}} = 200\text{M}$ under either the basic μP HPs (λ=0.1, η=0.01) or a full HP sweep (tuned). Inverting the fit at the target value yields $D_{\text{LR}}^{\text{equiv}}$, and N follows by division.

Reading the methodology figures. Each of Figures 20 and 21 has one panel per model scale (380M, 600M, 1.43B). Within a panel: the x-axis is Arabic corpus size D_{LR} (log scale), the y-axis is performance (accuracy in Fig. 20, Arabic VL in Fig. 21), and the solid curve through the markers is the log-linear fit to the monolingual sweep. Two horizontal lines mark the bilingual targets achieved at $D_{\text{LR}} = 200\text{M}$ under μP HPs (blue) and the full HP sweep (purple). To recover a multiplier from

one of those lines, follow it horizontally until it meets the monolingual curve, then drop down to read $D_{\text{LR}}^{\text{equiv}}$ on the x-axis (marked by a vertical dashed line); $N = D_{\text{LR}}^{\text{equiv}}/200\text{M}$. The further the intersection sits to the right of $D_{\text{LR}} = 200\text{M}$, the larger the multiplier.

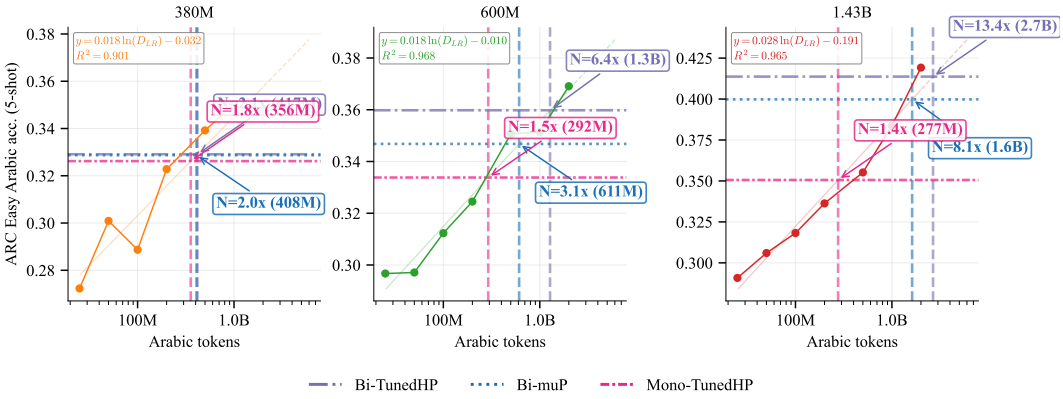


Figure 20: Data multiplier methodology (ARC Easy accuracy). Solid curves: monolingual accuracy vs. D_{LR}, with log-linear fit. Horizontal lines: bilingual performance at $D_{\text{LR}} = 200\text{M}$ under μP HPs (blue) and full HP sweep (purple). Vertical dashed lines: equivalent $D_{\text{LR}}^{\text{equiv}}$. The gap between the two targets grows with scale.

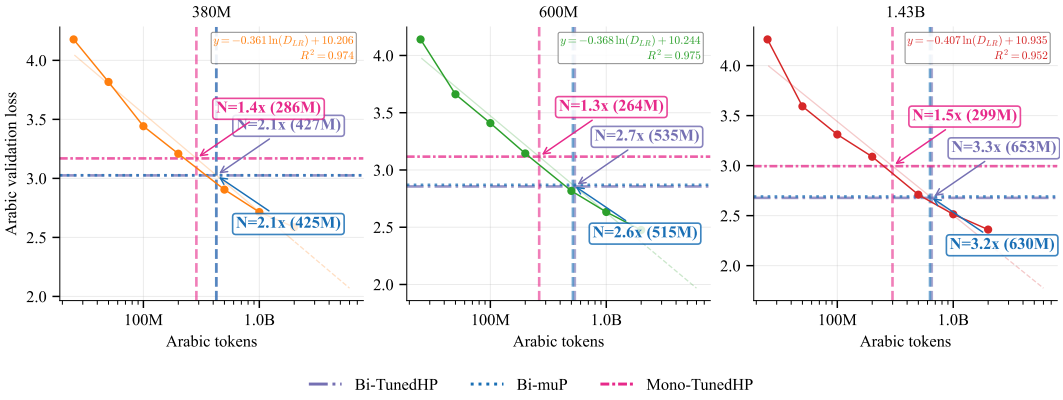


Figure 21: Data multiplier methodology (Arabic validation loss). Same format as Fig. 20. The VL multiplier stays in a narrow 2–3$\times$ band across scales, substantially smaller than the benchmark multiplier.

Results. On accuracy (Fig. 20), the horizontal gap between the bilingual targets and the monolingual reference grows steeply with scale, yielding N from 2$\times$ at 380M to 13$\times$ at 1.43B. On Arabic VL (Fig. 21), the gap stays narrow, yielding N in a 2–3$\times$ band across all scales. Table 14 summarises the accuracy multipliers under both controlled (identical HPs) and optimized (full HP sweep for bilingual) conditions. These two views are the basis of the divergence reported in §4.4.

Multiplier under validation-loss-based checkpoint selection. The main-text multiplier (Fig. 7) selects checkpoints by peak accuracy on the target benchmark, which requires running the benchmark at every checkpoint. A cheaper rule picks the min-VL checkpoint and reports accuracy there. Figure 22 recomputes the multiplier under this rule.

OBSERVATION. Under VL-based selection, the Bilingual-tuned-HP multiplier rises from 13.4$\times$ to 47.4$\times$ at 1.43B. The rise is driven by an asymmetry in how much accuracy each paradigm loses when forced to commit to min-VL. At 1.43B, the best-VL bilingual run loses 0.003 accuracy relative to its own peak-accuracy checkpoint (0.4137 $\rightarrow$ 0.4112); the monolingual run at $D_{\text{LR}} = 2\text{B}$ loses 0.039

Table 13: Log-linear fit coefficients for the data multiplier. Model: $y = a \cdot \ln(D_{\text{LR}}) + b$, where y is either ARC Easy accuracy or Arabic validation loss.

Model	Metric	a	b	r^2
380M	Accuracy	+0.018	−0.032	0.90
380M	Arabic VL	−0.361	+10.21	0.97
600M	Accuracy	+0.017	−0.006	0.97
600M	Arabic VL	−0.368	+10.24	0.98
1.43B	Accuracy	+0.027	−0.168	0.98
1.43B	Arabic VL	−0.407	+10.94	0.95

Table 14: Data multiplier N: unique Arabic data (as a multiple of $D_{\text{LR}} = 200$M) a monolingual model would need to match bilingual accuracy on ARC Easy (5-shot). Controlled: each model uses the base HPs that are best at 150M on its own training data, transferred with μP (λ=0.1, η=0.01 for bilingual; per corpus size for monolingual, Table 11). Optimized: bilingual gets a full HP sweep.

	Controlled		Optimized	
Model	Acc	N	Acc	N
380M	0.329	2.0×	0.329	2.1×
600M	0.347	3.1×	0.360	6.4×
1.43B	0.400	8.1×	0.414	13.4×

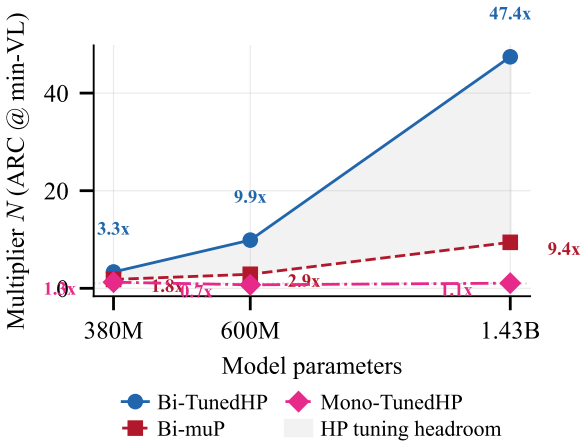


Figure 22: Data multiplier N vs. model size, with checkpoints selected by min Arabic VL rather than peak accuracy. Same paradigms as Fig. 7, right panel. The bilingual-tuned-HP multiplier inflates from 13.4× to 47.4× at 1.43B; the 1.43B value is an extrapolation of the log-linear fit 4.7× past the largest monolingual corpus ($D_{\text{LR}} = 2$B).

(0.4192 → 0.3805), an order of magnitude more. The same pattern holds across the monolingual sweep: accuracy at min-VL is 0.015–0.040 below peak at every D_{LR} and every scale, while bilingual gaps stay below 0.005.

EFFECT ON THE FIT. Switching from peak-accuracy to min-VL checkpoint selection shifts the two sides of the multiplier comparison asymmetrically. The bilingual target barely drops (best-VL is within 0.003 of best-accuracy for bilingual runs), while the monolingual reference curve drops by roughly 0.025 at every D_{LR} (monolingual runs lose much more accuracy when forced onto their min-VL checkpoint). With the target held roughly fixed and the reference curve pushed down, monolingual now needs far more data to reach the target, inflating N from 13.4× to 47.4× at 1.43B. At 1.43B, even $D_{\text{LR}} = 2$B on the shifted curve falls short of the target, so $D_{\text{LR}}^{\text{equiv}}$ is read from the log-linear fit extrapolated 4.7× past the fit range.

CAVEATS. The 4.7× extrapolation at 1.43B assumes the log-linear fit's slope holds past $D_{\text{LR}} = 2$B; we treat 13.4× (from peak-accuracy selection, Fig. 7) as the empirical lower bound and the VL-based numbers as upper-bound estimates contingent on fit extrapolation.

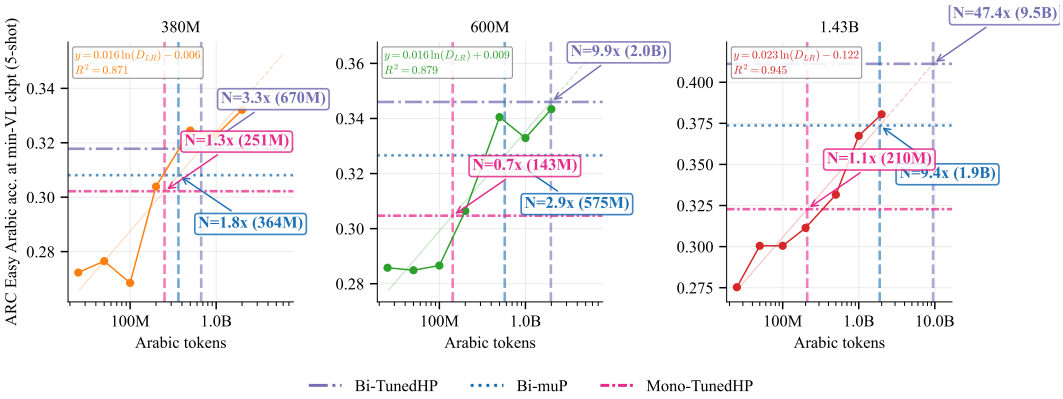


Figure 23: Data multiplier methodology under min-VL checkpoint selection (Arabic ARC Easy 5-shot). For each paradigm, accuracy is read at the checkpoint that minimizes Arabic VL rather than at peak accuracy. The monolingual reference curve shifts down relative to Fig. 20, while the bilingual targets barely move, producing the inflated multipliers in Fig. 22.

D.4 Data multiplier on SciQ and PIQA

To test whether the multiplier is specific to ARC Easy, we repeat the analysis on SciQ (Welbl et al., 2017) (884 questions) and PIQA (Bisk et al., 2019) (1838 questions), in Arabic and English, at 0-shot and 5-shot. SciQ is evaluated closed-book (question only) and open-book (question preceded by its supporting passage). Both are translated to Arabic with NLLB-200-3.3B, as for ARC Easy (§3).

Setup. We compare two configurations at $D_{\text{LR}} = 200\text{M}$: bilingual μP (λ=0.1, η=0.01, best over $R_{\text{max}} \in \{1, 2, 5, 10, 20, 50, 100\}$) and monolingual tuned (best over the 5×5 (λ, η) grid at $R_{\text{max}} = 100$). The bilingual full HP sweep was not evaluated on these benchmarks. The reference is a monolingual μP sweep over the seven corpus sizes of App. D.1, evaluated at every fourth saved checkpoint plus the last and the min-VL checkpoint. The multiplier follows App. D.3 with two differences: (i) the reference uses $(\lambda{=}0.1, \eta{=}0.01)$ and $R_{\text{max}} = 100$ at every corpus size rather than the per-corpus-size HPs of Table 11; (ii) the log-linear fit is constrained to pass through the monolingual point at $D_{\text{LR}} = 200\text{M}$, so $N = 1$ exactly when a configuration matches the monolingual run at 200M. At 380M, the $D_{\text{LR}} = 25\text{M}$ and 100M runs stopped saving checkpoints before their min-VL step; their peaks are lower bounds (hollow markers) and are excluded from the fit. We report multipliers for Arabic only: the monolingual models see no English, so an English multiplier is not meaningful.

Results. Figure 24 reports N; Figures 25–27 show the underlying peak accuracies. At 1.43B, bilingual μP exceeds monolingual tuning in all six settings (three benchmarks, two shot counts): 3.0–15.0$\times$ against 1.3–6.3$\times$, the same ordering as on ARC Easy (8.1$\times$ against $\sim$1.5$\times$). At 600M it leads in five of six; the exception, PIQA 0-shot, is a 0.4-point tie. At 380M neither dominates: monolingual tuning leads in four of six settings, by 0.2–1.3 points (within one standard error), and bilingual μP leads on open-book SciQ. The growth of N with scale replicates on PIQA ($2.0 \to 7.1 \to 15.0\times$ at 5-shot) but not on SciQ: closed-book stays at 1.3–4.7$\times$, comparable to the VL multiplier (2.1–3.2$\times$), and open-book peaks at 600M.

Mixing gains most where the benchmark needs the least stored knowledge. On open-book SciQ, where the supporting passage is in context, bilingual μP gains 9.2–13.5 points over the monolingual 200M run at 600M–1.43B; on closed-book SciQ it gains 0.6–4.3. This qualifies the account of §4.4: it suggests that part of what English contributes to Arabic benchmarks is reading comprehension and in-context skill, not only knowledge.

Caveats. (i) *Selection bias.* Peaks are maxima over noisy evaluations. The configurations pool more evaluations ($\sim$60–90 checkpoints for bilingual μP, $\sim$25 runs $\times$ 27 checkpoints for monolingual tuned) than each reference point (7–28 checkpoints), which inflates their peaks, most for monolingual tuning. (ii) *Sensitivity.* One standard error of accuracy is $\sim$1.6 points on SciQ and $\sim$1.2 on PIQA. The

PIQA reference curve is flat (0.7–1.1 points per e-fold of D_{LR}), so a one-standard-error shift changes N by a factor of 3–6; on SciQ the factor is 1.3–2.1. There are no seed replicates. (iii) *Extrapolation.* Four values imply $D_{\mathrm{LR}}^{\mathrm{equiv}} > 2\mathrm{B}$: SciQ open-book at 600M (13.4× 0-shot, 51.3× 5-shot) and 1.43B (11.9× 5-shot), and PIQA at 1.43B (15.0× 5-shot). The 51.3× value extrapolates 5× past the largest corpus; we read these as upper-bound estimates.

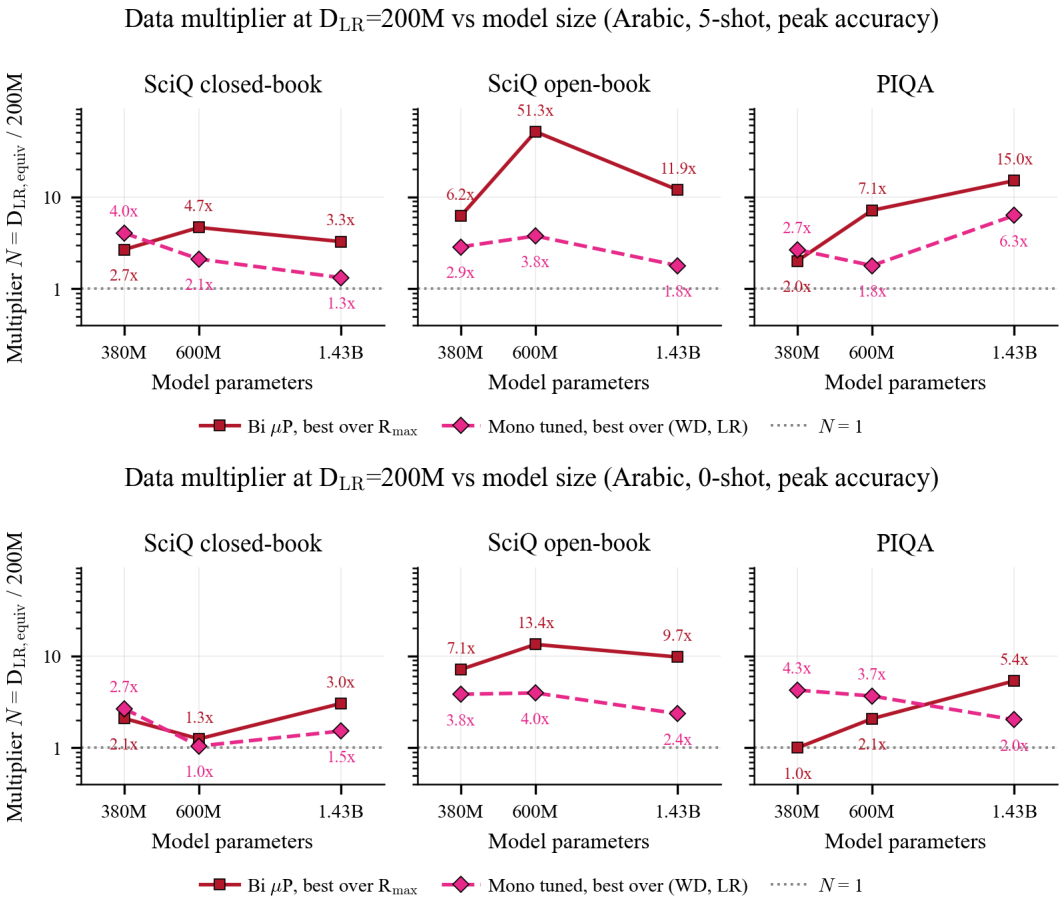


Figure 24: Data multiplier N vs. model size on SciQ and PIQA (Arabic, peak accuracy). **Top**: 5-shot. **Bottom**: 0-shot. Squares: bilingual μP, best over $R_{\max}$. Diamonds: monolingual, best over the (λ, η) grid. Dotted line: $N = 1$. At 1.43B, bilingual μP exceeds monolingual tuning on every benchmark.

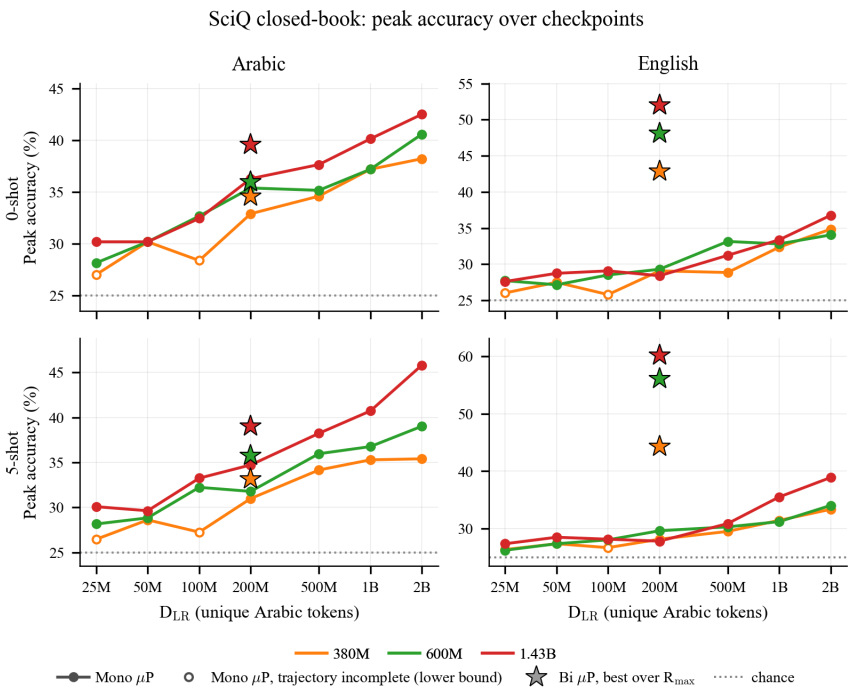


Figure 25: Peak SciQ closed-book accuracy vs. Arabic data budget (D_{LR}). Rows: 0-shot, 5-shot. Columns: Arabic, English. Lines: monolingual μP at each model scale; hollow markers are lower bounds (incomplete checkpoint trajectory). Stars: bilingual μP at $D_{\text{LR}} = 200\text{M}$, best over $R_{\max}$. Dotted line: chance.

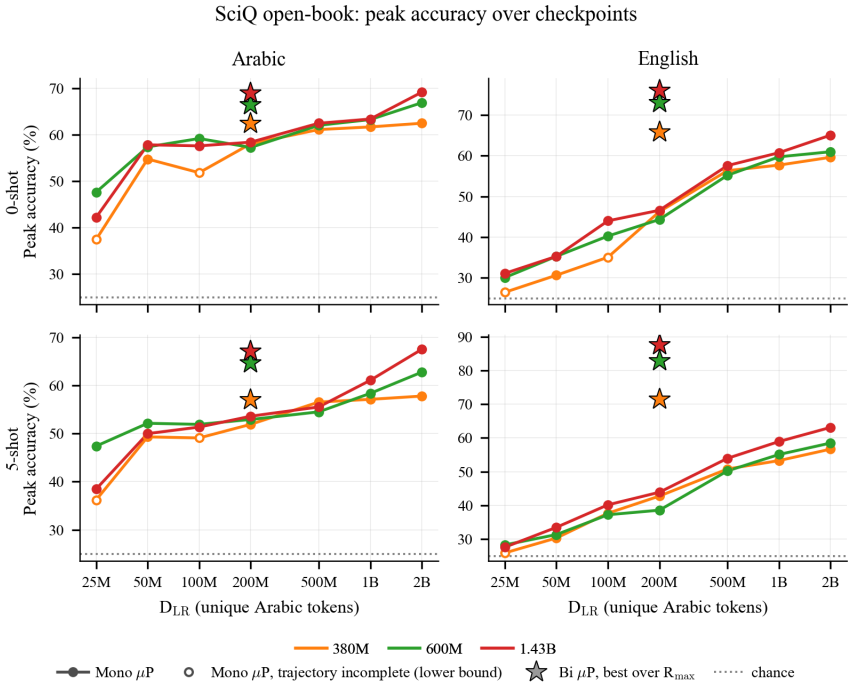


Figure 26: Peak SciQ open-book accuracy vs. D_{LR}. Same layout as Fig. 25.

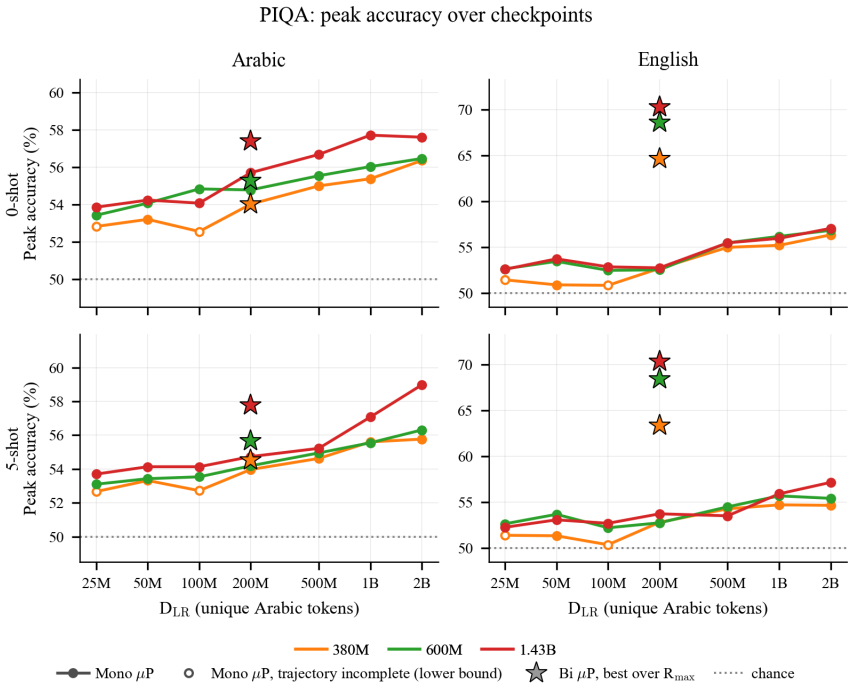


Figure 27: Peak PIQA accuracy vs. D_{LR}. Same layout as Fig. 25.

E ANOVA METHODOLOGY

Throughout this paper, we use analysis of variance (ANOVA) to measure how much of the total spread in Arabic validation loss is attributable to each tuning axis. Variance-based HP sensitivity analysis has substantial precedent in the HPO literature (Hutter et al., 2014; Probst et al., 2018; Watanabe et al., 2023); we use classical ANOVA on a structured grid of trained models. Our bilingual grid measures Arabic validation loss at every combination of $R_{\text{max}} \in \{1, 2, 5, 10, 20, 50, 100\}$ and the 25 hyperparameter cells ($5\,\lambda \times 5\,\eta$). The headline decomposition in §4.3 is two-way: R_{max} against a single 25-level HP factor. The three-way decomposition in App. G splits the HP factor into λ and η with their interaction. We describe the two-way version below; the three-way extension is standard.

Setup. Suppose we have a grid of outcomes y_{rh} indexed by two factors: a row factor $r \in \{1, \ldots, R\}$ (e.g. R_{max} levels) and a column factor $h \in \{1, \ldots, H\}$ (e.g. (λ, η) combinations). Each cell (r, h) contains one observation: the best Arabic validation loss for that configuration. The grand mean is $\bar{y} = \frac{1}{N} \sum_{r,h} y_{rh}$, where N is the number of observed cells.

Decomposition. ANOVA asks: when we look at all the variation in this grid, how much comes from changing the row factor vs. changing the column factor? It decomposes the total sum of squared deviations from the grand mean into additive terms:

$$\text{SS}_{\text{total}} = \text{SS}_{\text{row}} + \text{SS}_{\text{col}} + \text{SS}_{\text{residual}} \tag{1}$$

Each term measures how much variation one factor explains:

- $\text{SS}_{\text{row}} = \sum_r n_r \, (\bar{y}_{r\cdot} - \bar{y})^2$, where $\bar{y}_{r\cdot}$ is the mean of row r and n_r is the number of observed cells in that row. This captures how much the outcome changes when we vary the row factor (e.g. R_{max}), averaging over all column settings.
- $\text{SS}_{\text{col}} = \sum_h n_h \, (\bar{y}_{\cdot h} - \bar{y})^2$, defined analogously for columns. This captures the effect of varying HPs, averaging over all data settings.
- $\text{SS}_{\text{residual}} = \text{SS}_{\text{total}} - \text{SS}_{\text{row}} - \text{SS}_{\text{col}}$, the leftover. In a design with one observation per cell (as ours), this confounds the true interaction between factors with observation noise.

We report each term as a percentage of SS_{total}. If a factor explains 80% of the total variance, then choosing the wrong level of that factor is far more costly than choosing the wrong level of a factor that explains 16%.

Interpretation and caveats. A high SS_{row} fraction means the row means are spread far apart relative to the overall spread, i.e., the outcome is sensitive to the row factor. Conversely, a low fraction means rows have similar means, so the outcome is insensitive to that factor regardless of the column setting.

Two caveats apply. First, because each cell contains a single training run (no replication), the residual term cannot separate true factor interaction from noise. In practice the two-way residual is 4–6% across 380M, 600M, and 1.43B, and the three-way residual stays below 7% (App. G), so this limitation does not affect our conclusions. Second, the decomposition depends on which factor levels are included in the grid. We address this by reporting three operational views around the per-scale optimum and the μP HP (§4.3 and App. F), which makes the conclusions robust to grid boundary choices.

F SENSITIVITY AROUND THE OPERATIONAL POINT: THE CHOICE OF GRID MATTERS

How sensitive is bilingual training performance to HP choice around the operational point? The answer depends on how we define "around", and the choice matters more than one might expect. We report three operational views of the same data and show that they tell complementary stories, all converging on the same practical recommendation. Views 2 and 3 diverge sharply at 1.43B between an optimum-centered and a μP-centered analysis; App. G investigates that divergence and the full-grid pattern that produces it.

Three views of operational sensitivity. We consider three ways to restrict the ANOVA to an operational region:

1. **Threshold-based (Lebesgue split).** Keep the top-$\gamma\%$ of grid cells by performance. This is the construction used by PED-ANOVA (Watanabe et al., 2023) and corresponds to a *post-hoc* sensitivity analysis: it answers "among the configurations that turned out to be good, which factor mattered most?" The kept cells can be scattered anywhere in HP space.
2. **Contiguous neighborhood around the per-scale optimum (Riemann split).** Pick the per-scale optimal HP and expand a contiguous sub-grid radially outward. This answers "in a neighborhood of the actual best HP, which factor matters most?" It is an idealized lens, since identifying the per-scale optimum requires the full grid we are trying to analyze.
3. **Contiguous neighborhood around μP (Riemann split).** Same as (2) but anchored at the μP HP from a small proxy. This is the *a-priori* practitioner view: a real practitioner has a μP guess and wants to know how sensitive performance is to HP variation around it.

The threshold-based and contiguous views answer different questions. Threshold-based requires loss values for every cell (post-hoc), while contiguous-around-μP requires only a starting point (a-priori). The two contiguous views (around optimum vs around μP) coincide when μP transfers to the per-scale optimum exactly, and diverge when it does not.

View 1: threshold-based re-centering. For each scale, we sort all HP configurations by their mean Arabic VL across $R_{\max}$ levels and keep those within a relative threshold τ of the best. We then run two-way ANOVA on the kept cells. Figure 28 shows the variance decomposition across thresholds.

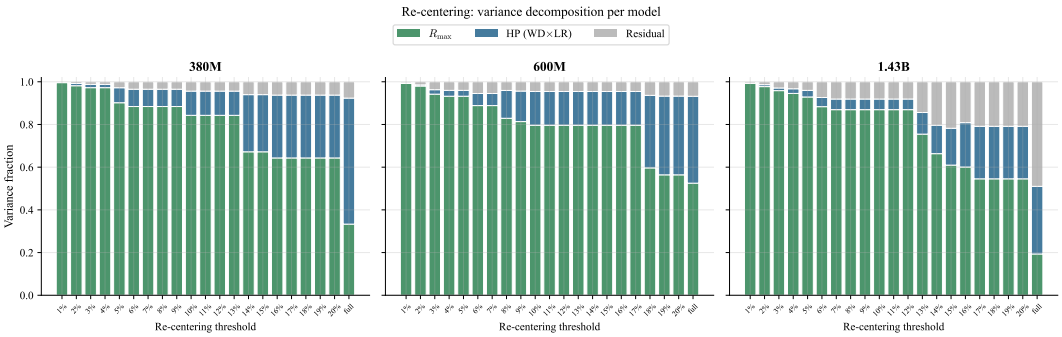


Figure 28: Threshold-based re-centering: two-way ANOVA variance shares vs. relative threshold τ (% above best HP mean loss). At every scale, $R_{\max}$ (green) dominates within reasonable thresholds. The full-grid HP-dominance at 1.43B (rightmost bar) disappears as soon as τ tightens.

The pattern is consistent across scales: $R_{\max}$ explains over 95% of variance at the tightest threshold ($\tau = 1\%$), 80–90% at $\tau = 10\%$, and remains the dominant factor up to $\tau \approx 14\%$. At 1.43B specifically, the full-grid pattern (HP dominance) reverses cleanly under any reasonable re-centering threshold. Within the operational region a practitioner who has done basic HP screening would occupy, mixing dominates.

The caveat is that this view assumes the practitioner can rank HPs by performance, which requires already having trained them all. It is informative for diagnosis but does not directly answer "what should I do before running the full grid."

Views 2 and 3: contiguous neighborhood around optimum vs μP. For each scale we pick a center HP (the per-scale optimum, or the μP HP) and expand a contiguous sub-grid radially: the r-th sub-grid contains $(r{+}1) \times (r{+}1)$ HP cells centered on the chosen point. We run two-way ANOVA on each sub-grid. Figure 30 shows the result.

At 380M and 600M, the two centers tell the same story: at small radii ($\leq 3{\times}3$), $R_{\max}$ explains over 90% of variance; HP variance only catches up as the sub-grid widens to include increasingly suboptimal configurations. The HP landscape is locally flat around both centers.

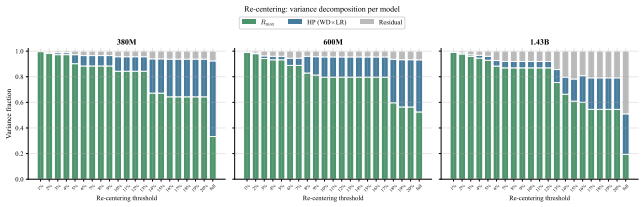


Figure 29: Companion view to Fig. 28: $R_{\max}$ ANOVA share as a function of the threshold τ (solid lines, left axis), with the number of HP configurations retained at each threshold overlaid (dashed lines, right axis). All three curves start near 100% at $\tau = 1\%$. 1.43B (red) tracks 380M and 600M closely up to $\tau \approx 12\%$, then drops sharply between $\tau = 12\%$ ($R_{\max} = 87\%$, 13/25 HPs) and $\tau = 14\%$ ($R_{\max} = 52\%$, 15/25 HPs): two configurations entering the grid at 13–14% above optimal are enough to cut $R_{\max}$'s share substantially. Used by App. G to argue that the 1.43B sharpening is broad rather than outlier-driven.

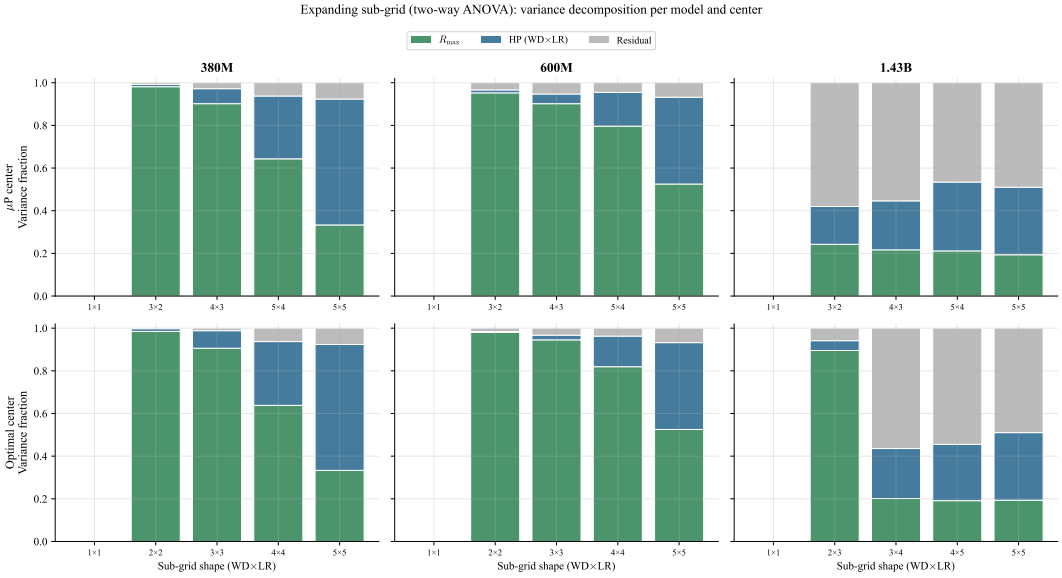


Figure 30: Contiguous neighborhood re-centering: two-way ANOVA variance shares as the sub-grid expands from the chosen center. *Top row*: μP center. *Bottom row*: per-scale optimum center. At 380M and 600M, the two centers give similar pictures: $R_{\max}$ dominates at small radii. At 1.43B, the two centers diverge sharply: optimum-centered (bottom right) shows clean $R_{\max}$ dominance at radius 1, but μP-centered (top right) shows mixed $R_{\max}$/HP variance and a large residual ($\sim$50%) at every radius.

At 1.43B, the two centers diverge sharply. Centered on the per-scale optimum ($\lambda = 0.01$, $\eta = 0.003$), the smallest sub-grid (radius 1, 2×3 cells) shows $R_{\max} = 89\%$, HP $= 5\%$. Centered on the μP HP ($\lambda = 0.1$, $\eta = 0.01$), the smallest sub-grid shows $R_{\max} = 24\%$, HP $= 18\%$, with a large residual ($\sim$50%). The three-way decomposition (Fig. 31) confirms that the residual mass is concentrated in the $\lambda \times \eta$ and $R_{\max} \times \lambda$ interactions, not in main effects.

Two findings emerge. First, at 380M and 600M, the μP HP and the per-scale optimum sit in the same locally flat basin: a practitioner using μP faces the same low HP sensitivity as a practitioner who has tuned. Second, at 1.43B, the two points have separated. The μP HP is in a region with substantially higher local HP sensitivity than the per-scale optimum. The basin around the actual optimum has shrunk below the precision of μP transfer.

What changes between 380M/600M and 1.43B that pushes the μP point outside the per-scale optimum's basin? The three-way decomposition of the μP-centered sub-grid above already hints at the answer: the residual mass is dominated by $\lambda \times \eta$ and $R_{\max} \times \lambda$ interactions that are absent at smaller scales. Appendix G investigates this mechanism on the full grid and shows that what looks

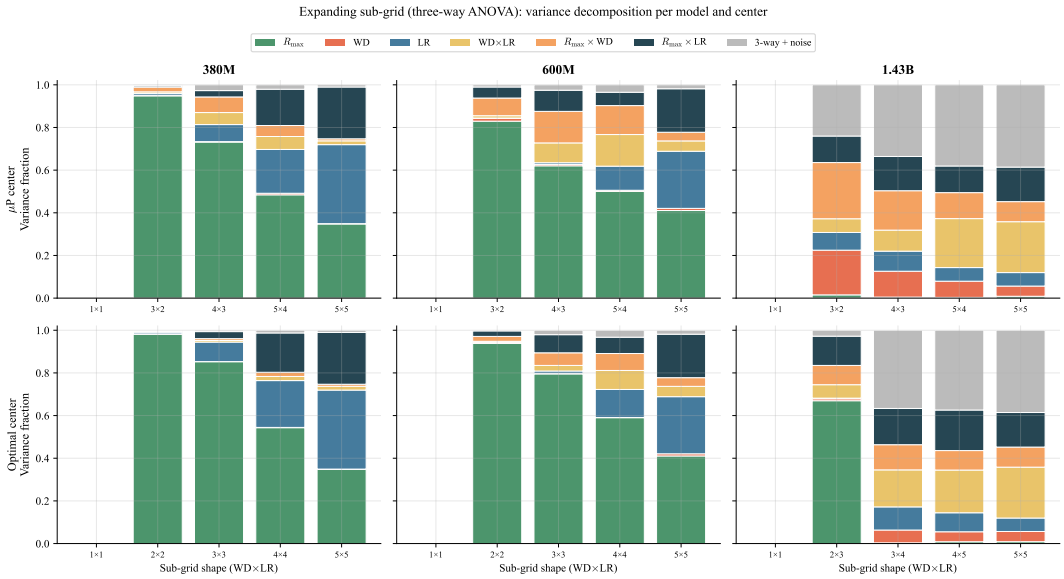


Figure 31: Same as Fig. 30 but with the three-way decomposition. At 1.43B, the μP-centered view (top right) is dominated by $R_{\max} \times$ LR and WD $\times$ LR interactions; the optimum-centered view (bottom right) is cleaner.

like "HP variance" at 1.43B is actually interaction variance: the loss landscape develops a coupled (λ, η) structure as scale grows.

Why μP remains the right default in all three views. The three views give different answers about how much HP sensitivity matters at 1.43B (low, low-but-cleaner, and substantial), but they converge on the same recommendation:

- If HP sensitivity is low (380M, 600M, or threshold-based view at any scale), μP HPs are within the flat basin, so per-scale tuning yields negligible improvement. Compute is better spent on mixing-ratio search.
- If HP sensitivity is high (1.43B, μP-centered view), the local landscape is jagged. The per-scale optimum is reachable but not by any cheap tuning protocol: a wide grid search at 1.43B exposes the practitioner to large variance from interaction-dominated regions of HP space. μP provides a known-decent point at zero search cost; further tuning is a high-variance bet that may or may not pay off.

In both regimes, μP is the safe default. When HP does not matter, the choice is free; when HP matters, μP avoids a costly and risky search. Practitioners with extra compute at scale can invest in per-scale HP refinement, but the marginal return is uncertain compared to the well-characterized return on mixing-ratio tuning.

A note on grid design. Our 5×5 HP grid was fixed across all model scales: $\lambda \in \{0.01, 0.1, 0.3, 1.0, 3.0\}$ and $\eta \in \{0.0001, 0.0003, 0.001, 0.003, 0.01\}$. Figure 32 tests whether the per-scale optimal HPs match the μP prediction, separately for LR and weight decay. μP (Yang et al., 2022) defines base HPs at a reference scale (150M in our setup, $d_{\text{model}} = 512$) and a width multiplier $m_N = d_{\text{model}}(N)/512$; the per-layer LR at scale N is η_{base}/m_N and the per-layer weight decay is $\lambda_{\text{base}} \cdot m_N$. The central promise is that the base HPs are scale-invariant, so tuning once at 150M transfers unchanged. Our grid values are the base HPs (pre-μP); μP applies the rescaling internally during training. For each HP we plot the nominal optimum (the best base grid value) and the effective optimum (the per-layer HP actually applied). Under perfect μP, the nominal curve is flat and the effective curve follows the μP scaling (η_{base}/m_N for LR, $\lambda_{\text{base}} \cdot m_N$ for WD).

For weight decay, μP transfers perfectly at every scale (Fig. 32, bottom row): the nominal optimum $\lambda^\star$ is flat at 10^{-2} across 380M, 600M, and 1.43B, and the effective $\lambda^\star \cdot m_N$ sits exactly on the μP

reference curve $10^{-2} \cdot m_N$ at every scale. One caveat, $\lambda = 0.01$ is the smallest value in our grid, so the true optimum could lie below it and we would not see it; the flatness we observe is consistent with perfect μP transfer but does not uniquely identify it. For LR (Fig. 32, top row), μP holds at 380M and 600M: nominal $\eta^\star = 10^{-2}$ at both scales, and the effective sits on the μP reference curve η_{base}/m_N. At 1.43B the nominal drops to $\eta^\star = 3 \times 10^{-3}$, a $\sim 3\times$ shift, and the effective correspondingly falls below the μP curve. The μP failure at 1.43B is therefore LR-specific, not a general breakdown of μP transfer.

The LR-specific drift matches the downward shift of the marginal η minimum in Fig. 33 (top-left), which reaches $\eta \approx 10^{-3}$ at 1.43B under a different aggregation, so the effect is robust to aggregation choice and is not a grid-noise artifact. One consequence is that the per-scale optimum frequently lands on grid boundaries: $\eta = 0.01$ (the largest grid value) at smaller scales, and $\lambda = 0.01$ (the smallest grid value) at every scale.

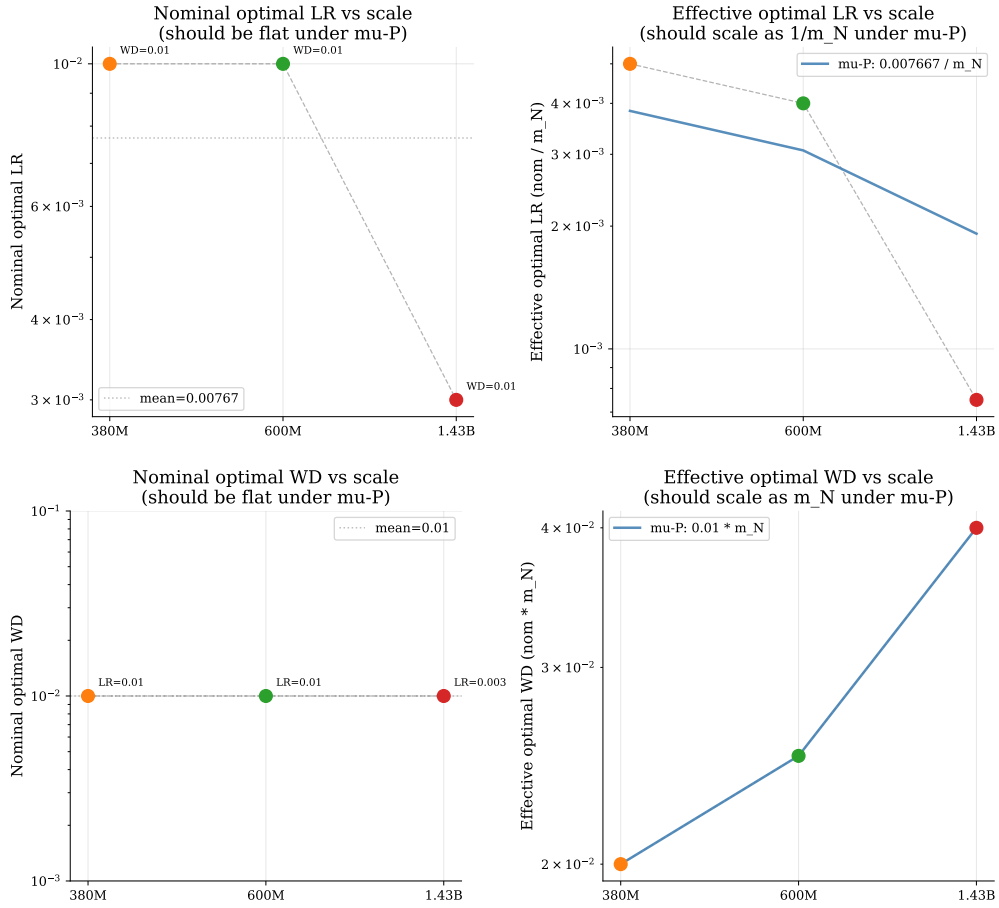


Figure 32: μP transfer test for LR (top) and weight decay (bottom), both as nominal (left column) vs effective (right column). *Top row, LR*: nominal $\eta^\star$ is the best base grid value; effective is $\eta^\star/m_N$. Under perfect μP the nominal is flat and the effective scales as η_{base}/m_N (blue reference curve). Observed: $\eta^\star = 10^{-2}$ at 380M and 600M (on curve), 3×10^{-3} at 1.43B (below curve), so μP fails at 1.43B for LR. *Bottom row, WD*: nominal $\lambda^\star$ is flat at 10^{-2} at every scale, and the effective $\lambda^\star \cdot m_N$ sits exactly on the μP reference curve $10^{-2} \cdot m_N$ at 380M, 600M, and 1.43B, so μP transfer for weight decay holds at every scale. The μP failure at 1.43B is therefore LR-specific. Caveat: $\lambda = 0.01$ is the smallest grid value, so the apparent flatness is consistent with but not uniquely diagnostic of perfect transfer.

The high-level conclusion is unaffected: practitioners should default to μP, and the mixing-vs-HP comparison in §4.3 remains valid because both axes are measured on the same grid.

Table 15: Number of HP configurations (out of 25) within a relative threshold of the best. Higher = flatter landscape.

Model	1%	2%	5%	10%	15%	20%
380M	2	6	9	15	18	20
600M	3	5	12	20	20	24
1.43B	2	4	10	13	16	21

Table 16: HP axis range as % of the best marginal mean loss. At 380M and 600M, η dominates and λ is negligible. At 1.43B, both axes become sensitive: η rises back to 27% and λ jumps from ≈4% to 20%.

Model	η (%)	λ (%)	η share
380M	32.4	4.0	89%
600M	16.1	3.9	81%
1.43B	27.3	19.9	58%

G HP LANDSCAPE SHARPENING AT SCALE

Appendix F and the main-body HP landscape figure (Fig. 5) show the same pattern: at 380M and 600M the HP landscape is broadly flat, and the μP HP sits in the same wide low-loss basin as the per-scale optimum. At 1.43B the near-optimal basin shrinks around the per-scale optimum. μP still reaches near-optimal loss (Table 18). This section investigates what drives the narrowing. The short answer: at 1.43B, pairwise couplings across $R_{\max}$, λ, and η become material on the full grid, and an additional three-way residual, as we show below, is largely concentrated in a handful of pathological cells. Main effects (the sum of the individual contributions of $R_{\max}$, λ, and η taken one at a time) explain only 12% of full-grid variance, versus 84% at 380M. Pairwise interactions add 50% spread across all three pairs, and a three-way residual of 39% (essentially absent at smaller scales) captures joint structure across $R_{\max}$, λ, and η that no single pair can explain. We walk through the evidence below.

Appendix F already decomposed the HP contribution at one specific anchor: the μP-centered neighborhood at 1.43B, where $\lambda \times \eta$ and $R_{\max} \times \lambda$ interactions dominated. This section extends that analysis to the full grid at each scale, so we can track how each piece shifts as the landscape narrows: the marginal effect of λ, the marginal effect of η, and the $\lambda \times \eta$ interaction. We first quantify the narrowing itself, then decompose the HP factor axis by axis, then separate main effects from the interaction with a three-way ANOVA, and finally check that the result is not an artifact of a few pathological cells that inflate $\mathrm{SS}_{\mathrm{HP}}$.

Measuring landscape flatness. For each model scale, we compute the mean Arabic validation loss of each (λ, η) combination across all $R_{\max}$ levels where it has data. Let ℓ^* denote the best such mean. For a relative threshold τ (expressed as a percentage), we count the number of HP configurations h satisfying $\bar{\ell}_h \leq \ell^* \cdot (1 + \tau)$. A flat landscape has many configurations near the optimum; a sharp one has few.

Table 15 and Fig. 29 (dashed lines, right axis) show the same pattern. We say the landscape *sharpens* with scale when fewer HP configurations survive a given threshold τ: the near-optimal basin holds fewer candidates as the model grows. By this measure, 1.43B sharpens against 600M at every threshold we tested. The comparison with 380M is tighter, with the two scales competitive at the tightest and loosest thresholds, so the narrowing is clearest against 600M. We turn next to which HP axis drives it.

Which HP axis drives the sharpening? The HP landscape has two axes, λ and η. For each axis, the sharpening at 1.43B could come from one of two changes. (i) The marginal minimum along that axis drifts away from its μP base value, so μP no longer lands at the axis optimum. (ii) The *range* of the marginal loss along the axis (worst minus best value) grows, so deviating from the minimum costs more than it did at smaller scales. We check both quantities for both axes below.

The top panels of Fig. 33 report where each axis's marginal minimum sits. On η, the minimum drifts roughly an order of magnitude below η_{base} between 380M and 1.43B (consistent with the nominal optimum drop in Fig. 32); on λ, it stays pinned at the smallest grid value at every scale. The bottom-left panel and Table 16 report each axis's range: η's range is comparable across scales, so η is *not* becoming sharper; λ's range jumps roughly $5\times$ between 600M and 1.43B. The bottom-right

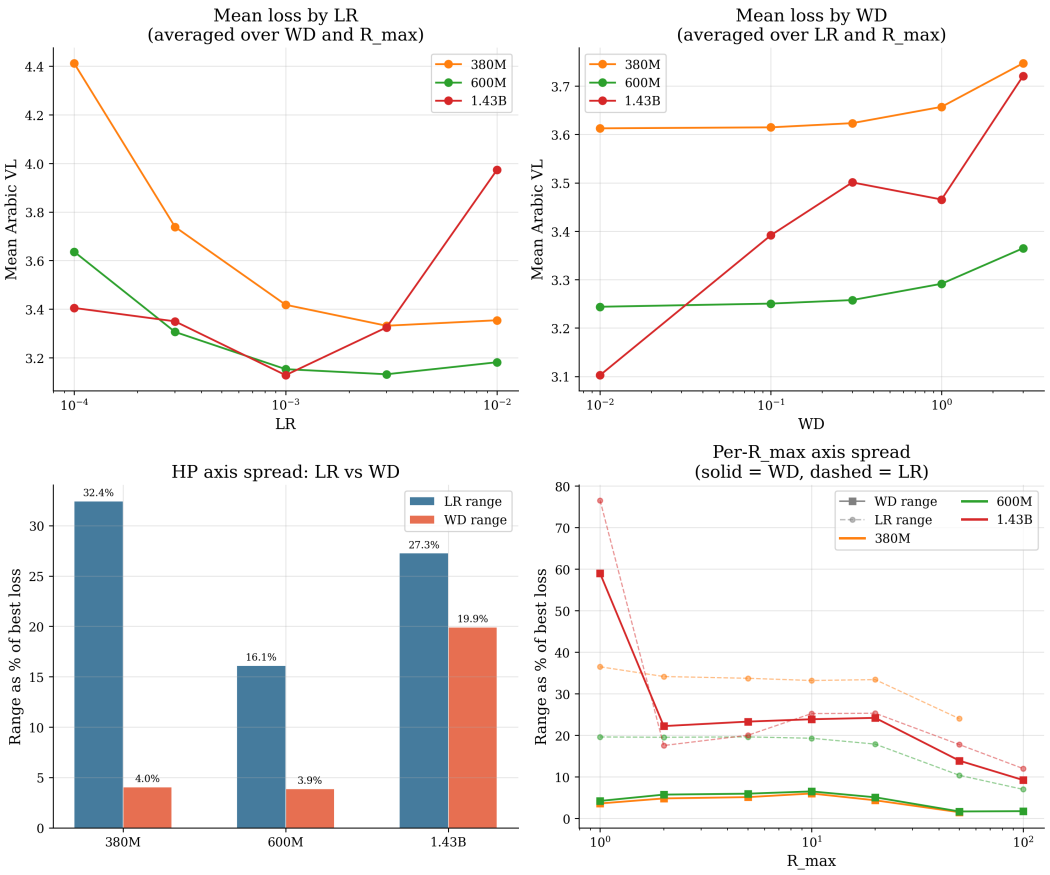


Figure 33: HP axis decomposition. Let $\bar{\ell}(\eta) = \text{mean}_{\lambda, R_{\text{max}}}\, \ell$ denote the mean Arabic VL along the η axis (averaged over λ and R_{max}), and $\bar{\ell}(\lambda)$ analogously. The *axis range* is $\text{Range}(\eta) = \big(\max_\eta \bar{\ell}(\eta) - \min_\eta \bar{\ell}(\eta)\big) / \min_\eta \bar{\ell}(\eta)$ (in %); $\text{Range}(\lambda)$ defined analogously. *Top row*: $\bar{\ell}(\eta)$ (left) and $\bar{\ell}(\lambda)$ (right) at each scale. *Bottom-left*: the two axis ranges per scale. A 20% range means the worst marginal along that axis is 20% above the best. *Bottom-right*: the same two ranges, but recomputed at each R_{max} separately (solid = λ, dashed = η); this checks whether the full-grid pattern holds conditional on R_{max}.

panel recomputes both ranges at each R_{max} separately and reproduces the same pattern, so the λ jump is not an averaging artifact.

Two distinct effects therefore drive the sharpening at 1.43B. On η, the minimum drifts away from η_{base}, so the μP choice sits off-optimum along that axis. On λ, the range widens, so weight-decay values that were indistinguishable at smaller scales now differ materially. The primary driver is λ; η's drift is a secondary contribution.

Three-way ANOVA: decomposing the HP factor. The two-way ANOVA treats (λ, η) as a single 25-level factor. This conflates the individual effects of λ and η with their interaction. To disentangle these, we run a three-way ANOVA with R_{max}, λ, and η as separate factors, including all two-way interactions. The residual captures the three-way interaction and noise.

Figure 34 and Table 17 show the results. At 380M, main effects explain 84% of variance, with η as the dominant factor (54%) and R_{max} second (30%). At 600M, R_{max} grows to 41% and η drops to 27%, but main effects still account for 69%. At 1.43B, the picture is qualitatively different. Main effects explain only 12% of variance. Pairwise interactions add 50%, distributed across $\lambda \times \eta$ (24%), $R_{\text{max}} \times \eta$ (16%), and $R_{\text{max}} \times \lambda$ (9%). The remaining 39% sits in the three-way residual, which at smaller scales was negligible (<2%).

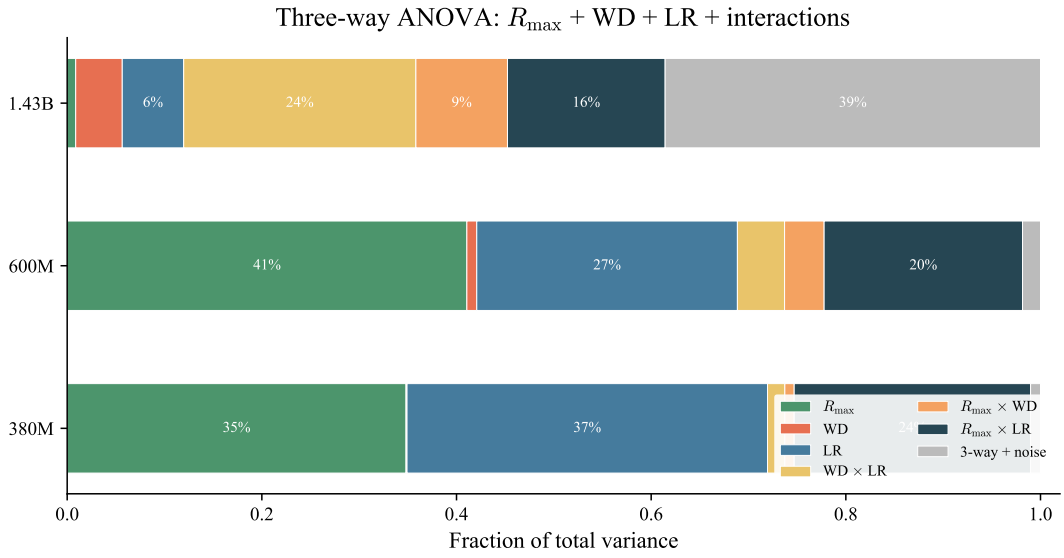


Figure 34: Three-way ANOVA decomposition of Arabic VL variance into main effects ($R_{\max}$, λ, η), two-way interactions ($\lambda \times \eta$, $R_{\max} \times \lambda$, $R_{\max} \times \eta$), and a three-way residual (three-way interaction plus noise). At 380M and 600M, main effects dominate (>69%). At 1.43B, main effects drop to 12%, pairwise interactions add 50% (spread across all three pairs), and a three-way residual of 39% captures joint structure that no single pair explains.

Table 17: Three-way ANOVA: fraction of Arabic VL variance explained by each main effect, pairwise interaction, and the three-way residual (three-way interaction plus noise). At 380M and 600M, main effects (especially $R_{\max}$ and η) dominate and the three-way residual is negligible. At 1.43B, no single term dominates: variance is spread across pairwise interactions (50% total) and a large three-way residual (39%).

Model	$R_{\max}$	λ	η	$\lambda \times \eta$	$R_{\max} \times \lambda$	$R_{\max} \times \eta$	Residual
380M	30.2%	0.1%	53.8%	2.9%	1.0%	11.0%	1.1%
600M	41.1%	1.0%	26.8%	4.8%	4.1%	20.4%	1.8%
1.43B	0.9%	4.8%	6.3%	23.9%	9.4%	16.2%	38.6%

The practical reading: on the full 1.43B grid, the loss landscape is no longer approximately additive in any single hyperparameter. At 380M and 600M, λ and η are approximately independent ($\lambda \times \eta < 5\%$ of variance), so a practitioner can tune them along separate 1D sweeps and compose the results. At 1.43B this factorisation breaks. The $\lambda \times \eta$ interaction alone carries 24% of variance, more than twice the sum of the individual λ and η main effects (11%). Concretely, the best λ depends on which η is chosen and vice versa: marginalising over one axis returns a coordinate that is suboptimal at the other. A practitioner who tunes η first at fixed λ, then λ at fixed η, can miss the joint (λ, η) optimum by several percent at this scale. The coupling extends to $R_{\max}$: the $R_{\max} \times \lambda$ and $R_{\max} \times \eta$ interactions add another 25% of variance, so the joint (λ, η) optimum itself shifts with the mixing ratio. The remaining 39% sits in the three-way residual, capturing variance no pair can predict; the next paragraph shows that this residual is largely concentrated in 5 outlier cells, so the bulk of the grid is pairwise-structured ($R_{\max} \times \lambda$ and $\lambda \times \eta$) rather than fully non-separable. In operational terms, per-scale tuning at 1.43B still couples all three axes: the $R_{\max} \times \lambda$ interaction means that the best $R_{\max}$ is not strictly independent of (λ, η). The μP recipe keeps this penalty small: transferring HPs from the 150M proxy and tuning $R_{\max}$ at those HPs lands within 0–3.4% of the per-scale optimum (Table 18), and the 0–3.4% residual is exactly what the $R_{\max} \times \lambda$ coupling predicts.

Is the 1.43B sharpening driven by a few pathological cells? The 39% three-way residual raises a natural question: is the non-separability spread across the grid, or concentrated in a few cells? We flag IQR outliers on the full grid: for each (λ, η) cell, we take its mean Arabic VL across all $R_{\max}$

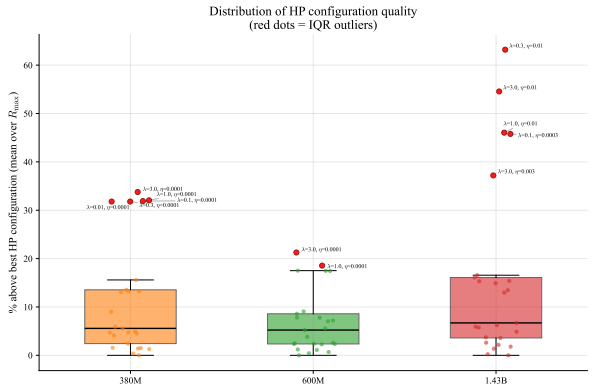


Figure 35: Distribution of HP configuration quality at each scale. Each point is one (λ, η) combination, measured as % above the best. Boxes show the IQR; red dots are statistical outliers (beyond $1.5 \times$ IQR).

levels as a percentage above the best configuration, and mark cells beyond $1.5\times$ IQR. Figure 35 shows the resulting quality distribution: 5 outliers at 380M, 2 at 600M, and 5 at 1.43B.

Figure 36 re-runs the ANOVA with and without those outliers, under both the two-way decomposition (R_{max} vs. the HP factor) and the three-way decomposition. The answer to the opening question splits along the two views.

Three-way: outlier-driven. The three-way residual at 1.43B collapses from 39% to ∼4% once the 5 cells are removed, and the rest of the grid becomes pairwise-dominated ($R_{\text{max}} \times \lambda$ and $\lambda \times \eta$). Mechanically, those 5 cells fail at particular $(\lambda, \eta, R_{\text{max}})$ triples rather than along any single axis or pair, and an ANOVA routes variance of that kind into the three-way residual, which is why removing them empties it.

Two-way: broad. The HP-factor share at 1.43B drops from 32% to 25% when the outliers are removed, not to zero. The HP-dominance flagged in the main body (Q1) is therefore a broad property of the 1.43B grid, not an artifact of a handful of cells. The threshold-based and optimum-centered re-centering views in App. F agree on independent definitions, and the μP cell $(\lambda{=}0.01, \eta{=}0.01)$ sits far from the outlier region at every scale, so the HP-dominance that survives outlier removal is what shapes the landscape around μP.

Conclusion: the narrowing is genuine, but μP transfer still works. This section built up a working definition of "narrowing" and tested it with independent views. The narrowing is the shrinking of the near-optimal basin: at a relative threshold τ, fewer HP configurations lie within τ% of the best. Table 15 shows this directly (1.43B has fewer configurations within the 2–15% thresholds than 380M or 600M); Fig. 36 shows the sharpening is not fully explained by outliers (the two-way HP share at 1.43B drops from 32% to 25% after removing the 5 IQR outliers, not to zero); Table 16 shows the λ range jumps $5\times$ between 600M and 1.43B while the η optimum drifts an order of magnitude below η_{base} (Fig. 32, Fig. 33); and the three-way ANOVA shows main effects dropping from 84% at 380M to 12% at 1.43B, with interactions and a three-way residual absorbing the rest (Table 17). The narrowing is therefore a genuine property of the 1.43B loss landscape, built from a λ-sensitivity jump, an η-optimum drift, and the emergence of a $\lambda \times \eta$ interaction.

The μP prediction that the optimum sits at a scale-invariant base LR does not hold strictly: the nominal optimum $\eta^\star$ drops from 10^{-2} at 380M and 600M to 3×10^{-3} at 1.43B (Fig. 32, left panel). What still holds is the weaker claim that matters in practice: the μP HP stays inside the near-optimal basin even as that basin shrinks and shifts. Table 18 shows the μP cell ($\lambda{=}0.01$, $\eta{=}0.01$) is within 0–3.4% of the per-R_{max} best at every scale, and Fig. 37 shows the same gap is below 0.6% on bilingual Arabic VL at all scales up to 1.43B, with ARC-Easy Arabic 5-shot accuracy tracking the validation-loss gap. Sharpening raises the cost of a *bad* HP choice (extreme λ or far-from-optimum η) but not the cost of the μP choice specifically. The practical risk at 1.43B is therefore that a wide grid search observes HP-dominated variance as far corners of the grid become pathological, not that μP transfer itself fails.

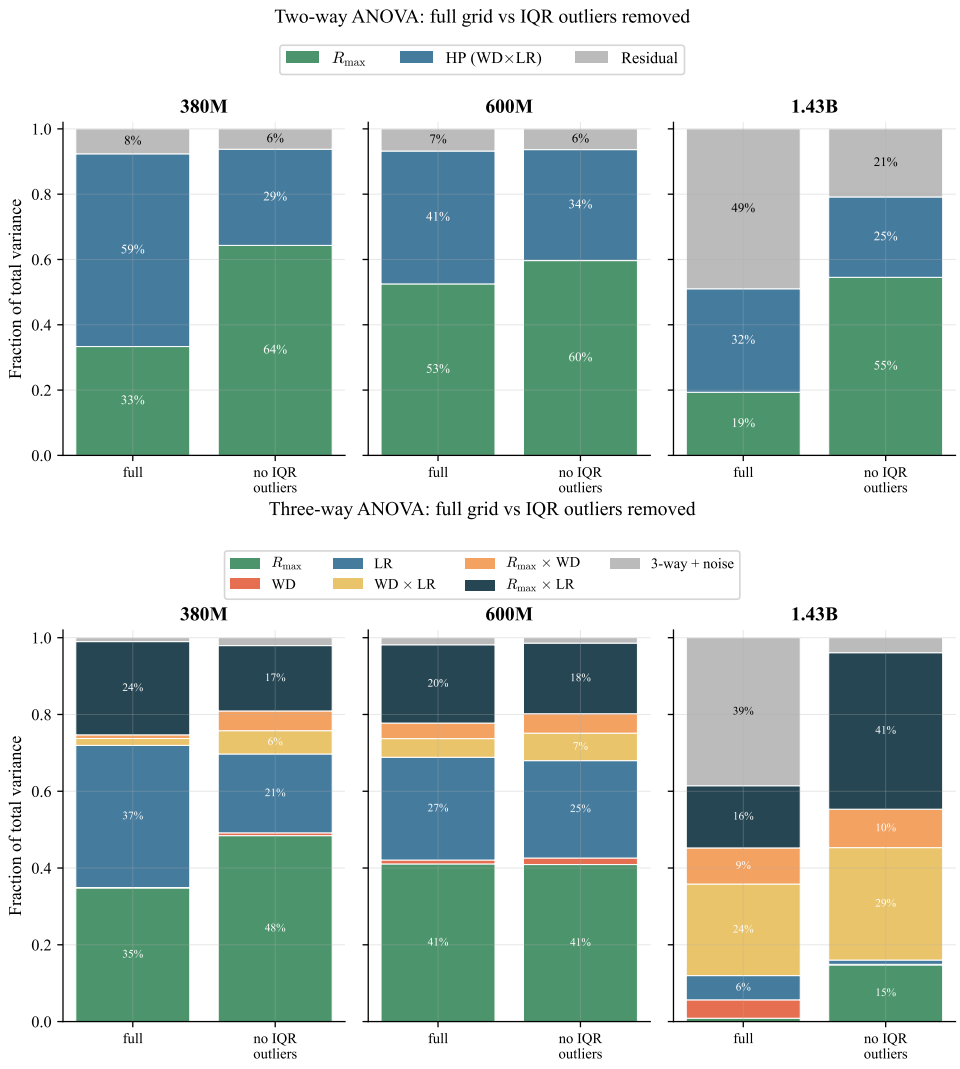


Figure 36: ANOVA decomposition of Arabic VL variance, full grid vs. IQR outliers removed. *Top*: two-way ANOVA treating (λ, η) as a single HP factor. *Bottom*: three-way ANOVA with $R_{\max}$, λ, η as separate factors plus all two-way interactions. Removing outliers flips the 380M picture entirely ($R_{\max}$ from 30% to 63% in two-way) and recovers $R_{\max}$ as the dominant factor at 1.43B (19% to 55%); the 1.43B three-way residual collapses from 39% to ∼4%, with what remains dominated by $R_{\max} \times \lambda$ (41%) and $\lambda \times \eta$ (29%).

The practical implication is twofold. First, μP hyperparameter transfer works: a practitioner who selects λ=0.01, η=0.01 at small scale and transfers it to 1.43B obtains near-optimal performance at every $R_{\max}$. Second, the risk at larger scales is not that the optimal HP shifts, but that the penalty for choosing a poor HP (especially high weight decay) grows. A practitioner who trusts the μP HP is safe; a practitioner who runs a wide grid search at 1.43B will observe HP-dominated variance because the far corners of the grid become increasingly pathological.

Table 18: Performance of the μP HP (λ=0.01, η=0.01) vs. the per-$R_{\max}$ best at each scale. Gap is expressed as % of the best loss. Even though the per-scale optimum drifts with scale (Fig. 32), the μP HP stays within a few percent of it; the cost of deviating from the μP cell grows with scale only at the grid extremes.

Model	$R_{\max}$	Best loss	λ=0.01, η=0.01	Gap (%)
380M	1	3.572	3.572	0.0
380M	10	3.097	3.097	0.0
380M	50	3.065	3.151	2.8
600M	1	3.396	3.396	0.0
600M	10	2.914	2.914	0.0
600M	50	2.897	2.962	2.2
600M	100	2.998	3.053	1.8
1.43B	5	2.973	3.007	1.1
1.43B	10	2.836	2.866	1.1
1.43B	20	2.749	2.749	0.0
1.43B	50	2.678	2.737	2.2
1.43B	100	2.681	2.774	3.4

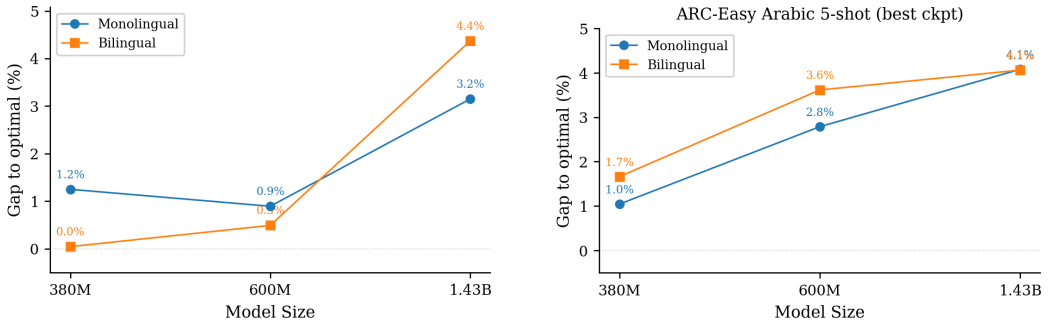


Figure 37: Relative gap between μP HPs and the per-scale optimum. **Left**: Arabic validation loss. **Right**: ARC-Easy Arabic 5-shot accuracy (best checkpoint). The benchmark gap tracks the validation loss gap, confirming that μP transfer generalizes to downstream performance. In the bilingual setting, the gap is $<$0.6% on VL at all scales up to 1.43B.

H Data quantity vs. hyperparameter variance decomposition

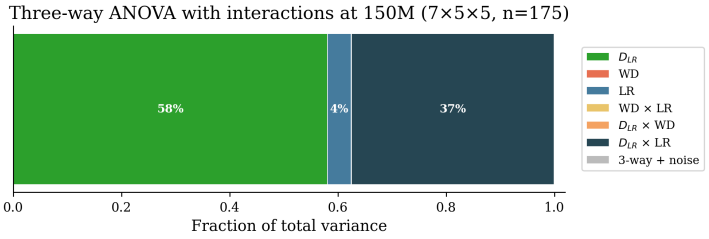
The main text shows that data mixing dominates HP tuning in the bilingual setting (§4.3, Q1). Here we ask the more direct question: does *raw unique data quantity* dominate HP choice in monolingual training? We answer with two complementary analyses: a three-way ANOVA at 150M with all two-way interactions and a re-centering robustness sweep, at the only scale where we have a full $D_{\text{LR}} \times \lambda \times \eta$ factorial grid; and an effect-size comparison across all four scales using the partial grids that exist at larger scales. Both follow the ANOVA framework described in App. E.

Three-way ANOVA at 150M. We run a balanced factorial experiment: 7 Arabic corpus sizes ($D_{\text{LR}} \in \{25\text{M}, \ldots, 2\text{B}\}$) $\times$ 5 weight decay values $\times$ 5 learning rates = 175 monolingual runs at 150M, each trained to 100 repetitions. We fit a full three-way ANOVA with main effects and all two-way interactions (Type III sums of squares) and decompose Arabic VL variance across the seven resulting terms. Table 19 reports the result.

D_{LR} alone explains 58% of Arabic VL variance, η explains 4%, and a large $D_{\text{LR}} \times \eta$ interaction contributes another 37%. The interaction is structured, not residual noise: the effect of η depends strongly on D_{LR}, with η=0.0001 severely undertraining the model at large D_{LR} (100 repetitions under a tiny LR) while being less harmful at small D_{LR}. Weight decay and its interactions contribute essentially nothing. Combining the main effect and its interaction with η, D_{LR} accounts for 96% of

Table 19: Three-way ANOVA at 150M on the complete $7 \times 5 \times 5$ grid ($n = 175$, no missing cells): Type III variance decomposition with main effects and all two-way interactions. *Left*: variance fractions (the four terms contributing $< 0.1\%$ — λ, $\lambda \times \eta$, $D_{\text{LR}} \times \lambda$, three-way+noise — are omitted). *Right*: same decomposition as a stacked bar.

Term	Variance fraction
D_{LR}	58.1%
$D_{\text{LR}} \times \eta$	37.5%
η	4.4%



the variance; everything involving η alone or λ sums to 4%. The bilingual ANOVA in §4.3 uses R_{max} (repetition count) on the data axis rather than unique Arabic tokens, and is not reported at 150M. With unique Arabic tokens as the data axis, data dominates already at 150M.

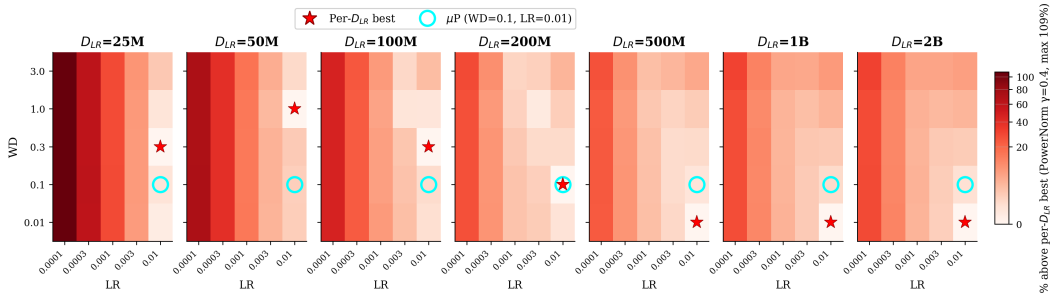


Figure 38: HP landscape at 150M, faceted by Arabic corpus size D_{LR}. Each panel is the 5×5 (λ, η) grid at one D_{LR}; cell color is % above the per-D_{LR} best (PowerNorm γ=0.4, capped at 100%). Red star: per-D_{LR} best. Cyan circle: μP HP (λ=0.1, η=0.01). The per-D_{LR} best λ tracks repetition pressure: high λ at small D_{LR} (heavy repetition), dropping to the smallest grid value λ=0.01 once $D_{\text{LR}} \geq 500$M. The dark η=0.0001 column at every D_{LR} is the $D_{\text{LR}} \times \eta$ interaction the ANOVA reports.

Re-centering sweep. The $D_{\text{LR}} \times \eta$ interaction concentrates in pathological corners of the HP grid (mainly η=0.0001 paired with large D_{LR}). We verify D_{LR} dominance via a re-centering sweep: for each threshold $T \in \{3\%, \ldots, 20\%\}$, retain only the HPs whose marginal mean Arabic VL (averaged over D_{LR}) is within $T\%$ of the best HP, then refit the three-way ANOVA on the filtered subgrid. Figure 40 plots the decomposition as a function of T.

D_{LR} dominance is stable across the sweep: 94–96% at tight thresholds ($T \leq 9\%$, retaining 8–10 of 25 HPs near the per-axis optimum), 80% at moderate thresholds ($T \geq 11\%$, retaining 15 of 25 HPs), and 58% on the full grid. The complement is the $D_{\text{LR}} \times \eta$ interaction, which grows as more pathological η settings re-enter the subgrid. The T=10% point is a balanced-factorial artifact rather than a finding: the cutoff retains 12 of 25 HPs but only 2 of 5 weight decays at η=0.001, so the filtered subgrid is a staircase rather than a rectangle, and Type III SS cannot cleanly separate λ from η under this geometry (spurious λ and $\lambda \times \eta$ mass appears). At T=11% the subgrid becomes rectangular again and the decomposition returns to $D_{\text{LR}} = 80\%$.

Effect-size comparison across scales. At 380M, 600M, and 1.43B we lack a full $D_{\text{LR}} \times \lambda \times \eta$ grid, so a three-way ANOVA is not available. We substitute a model-free proxy for variance dominance: the ratio of the Arabic VL range along the data axis to the Arabic VL range along the HP axis. Let $\ell(D, \lambda, \eta; N)$ denote the best Arabic validation loss for a model of size N trained on D unique Arabic tokens with weight decay λ and learning rate η, and let $\mathcal{H}_T(N)$ denote the set of HPs (λ, η) whose marginal mean VL (averaged over D) lies within a fraction T of the best HP's marginal mean,

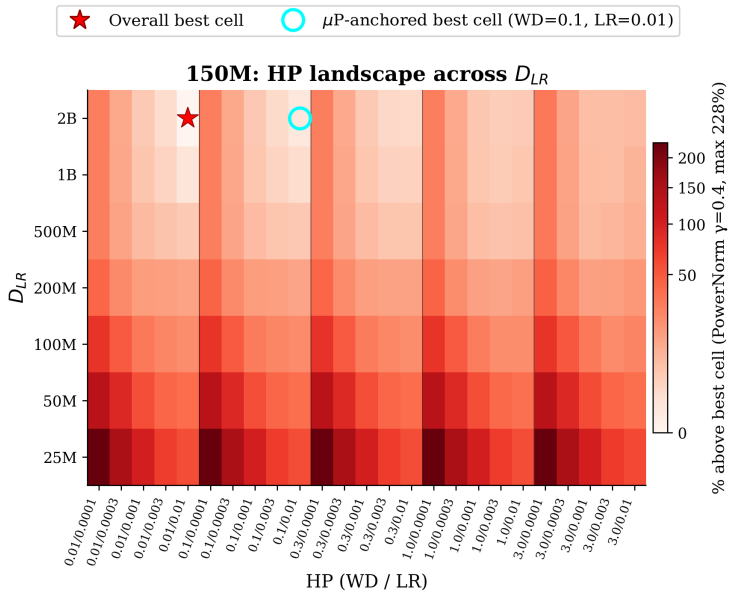


Figure 39: Same data as Fig. 38, arranged as a single heatmap: D_{LR} on the y-axis, the 25 (λ, η) HP cells on the x-axis. Cell color is % above the overall best. Red star: overall best at D_{LR}=2B, λ=0.01, η=0.01. Cyan circle: μP-anchored best at λ=0.1, η=0.01, also at D_{LR}=2B.

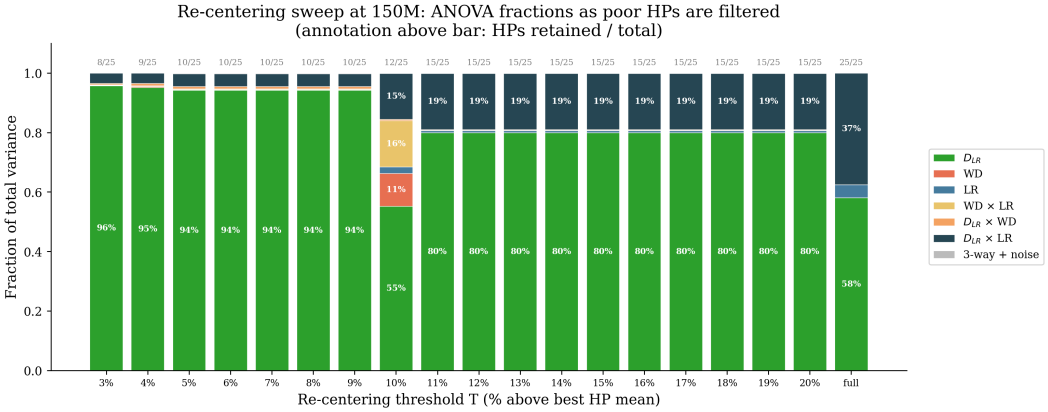


Figure 40: Re-centering sweep at 150M: three-way ANOVA variance fractions as poorly-performing HPs are filtered out. At each threshold T we retain HPs whose marginal mean Arabic VL (averaged over D_{LR}) is within $T\%$ of the best HP; the right-most "full" bar is the unfiltered $7 \times 5 \times 5$ grid. Annotations above each bar: retained HPs / 25 total. D_{LR} dominates at every threshold (55–96%). The T=10% bar is a balanced-grid artifact (see text), not a substantive finding.

matching the re-centering used for the ANOVA above. Define

$$\text{Range}_D(N) = \max_D \ell\big(D, \lambda^\star(D), \eta^\star(D); N\big) - \min_D \ell\big(D, \lambda^\star(D), \eta^\star(D); N\big), \tag{2}$$

$$\text{Range}_{\text{HP}}(N; T) = \max_{(\lambda,\eta)\in\mathcal{H}_T(N)} \ell(200\text{M}, \lambda, \eta; N) - \min_{(\lambda,\eta)\in\mathcal{H}_T(N)} \ell(200\text{M}, \lambda, \eta; N), \tag{3}$$

where $D \in \{25\text{M}, 50\text{M}, 100\text{M}, 200\text{M}, 500\text{M}, 1\text{B}, 2\text{B}\}$ with per-size tuned HP $(\lambda^\star(D), \eta^\star(D))$ and (λ, η) ranges over the 5×5 grid at fixed $D = 200\text{M}$. Taking $T = \infty$ recovers the full HP grid (no filtering). The data-axis dominance ratio

$$\rho(N; T) = \frac{\text{Range}_D(N)}{\text{Range}_{\text{HP}}(N; T)} \tag{4}$$

compares the two effect sizes: $\rho > 1$ means sweeping the data axis moves Arabic VL more than sweeping the (post-thresholding) HP axis. Two asymmetries bias ρ downward, against our claim: (i) with up to 25 cells on the HP side vs. 7 on the data side, the HP range has more opportunity to reach an extreme loss, and (ii) the data-axis HPs are retuned per D, keeping every point near its local optimum and compressing Range_D. Any observed $\rho > 1$ is therefore a conservative lower bound on the true data-axis dominance.

Table 20: Data-axis dominance ratio $\rho(N;T)$ at each model scale, on the full HP grid and after re-centering at $T \in \{10\%, 20\%\}$; n_{HP} is the number of HPs retained out of 25. On the full grid the ratio drops at 1.43B; after re-centering, dominance is stable at $\sim 5\times$ and is largest at 1.43B.

			$\rho(N;T)$		
Model	Range_D	$\mathrm{Range}_{\mathrm{HP}}$ (full)	Full grid	T=10% (n_{HP})	T=20% (n_{HP})
150M	1.71	0.96	1.8×	5.1× (20)	5.1× (20)
380M	1.59	0.45	3.6×	5.1× (22)	3.6× (25)
600M	1.66	0.31	5.4×	5.4× (25)	5.4× (25)
1.43B	1.90	0.97	2.0×	6.7× (15)	5.0× (24)

On the full grid, ρ rises from $1.8\times$ at 150M to $5.4\times$ at 600M and then drops to $2.0\times$ at 1.43B. After re-centering, the drop at 1.43B disappears: at $T = 10\%$, ρ is in the range 5.1–6.7× at every scale and is in fact *largest* at 1.43B (6.7×), reaching $15.1\times$ at $T = 5\%$. The full-grid drop was therefore driven by HP pathologies specific to 1.43B, where only 15 of 25 HPs survive a 10% cutoff (vs. 20–25 at smaller scales): the HP range is inflated by a small number of bad cells rather than by a broadly sharper landscape. Data-axis dominance at 1.43B is not a counterexample to the scaling trend but a consequence of the same pathology flagged in §4.3 (Q3). Figure 41 plots $\rho(N;T)$ for $T \in \{5\%, 10\%, 20\%, \infty\}$.

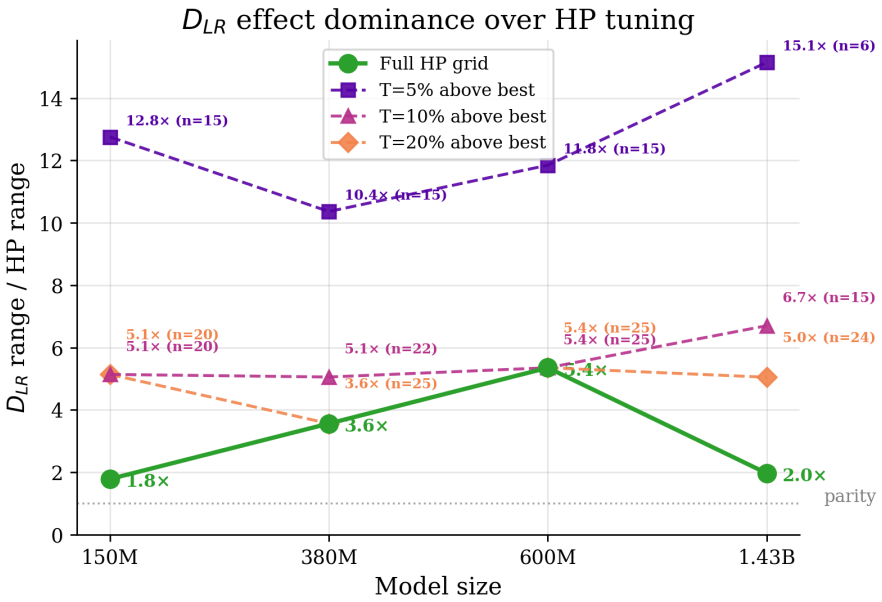


Figure 41: Data-axis dominance ratio $\rho(N;T) = \mathrm{Range}_D(N)/\mathrm{Range}_{\mathrm{HP}}(N;T)$ by model scale (Eqs. 2–4). Green solid ($T = \infty$): full HP grid. Purple, pink, orange (dashed): HP range recomputed after retaining HPs within $T \in \{5\%, 10\%, 20\%\}$ of the best. Annotations: ρ value and n_{HP}. The full-grid dip at 1.43B is absent from every re-centered curve; once pathological HPs are filtered, ρ is stable ($\sim 5\times$) across scales and largest at 1.43B, consistent with 1.43B having the sharpest HP landscape rather than the weakest data effect.

I CONCURRENT WORK

Sedova et al. (2026) study mixture pre-training under data constraints, deriving a repetition-aware scaling law that predicts target-domain loss as a joint function of model size, total compute, target-pool size, and mixture weight, fitted at small scales and extrapolated to larger ones. They validate the law across more than 2000 runs spanning multilingual (German, French, Swahili), multi-domain (mathematics, scientific papers, Wikipedia), and quality-filtered settings at 101M–936M parameters, and use it to recover optimal repetition counts that reach 15–20 for the constrained source. Our paper addresses the prior question: comparing data mixing against hyperparameter tuning (weight decay, learning rate) as competing strategies for data-constrained pre-training, and quantifying mixing's advantage over tuning in equivalent unique target tokens across model scales. We further show that target-language validation loss systematically underestimates mixing's downstream value: the gap between validation loss and benchmark accuracy widens with scale, with mixing yielding the equivalent of 2–3$\times$ more unique target data on validation loss but 2–13$\times$ on downstream task accuracy.

J LIMITATIONS

Mixing adds information that tuning cannot. At a fixed Arabic budget, mixing introduces English tokens while hyperparameter tuning only re-weights repeated Arabic, so the two levers are not matched on information. This is deliberate, since it is the choice a practitioner faces when Arabic is the scarce resource, and §4.4 controls for it by pricing mixing against monolingual models trained on more unique Arabic. What our design does not separate is how much of mixing's benefit comes from the diversity of a second language and how much from the content English carries. The clean test is to mix in an English translation of the same 200M Arabic tokens, holding content fixed and varying only the language. We did not run it: producing the translated corpus and retraining the grid is a study in itself, and we leave it to future work.

Benchmark coverage. Our main downstream analysis rests on ARC Easy, evaluated in both English and Arabic. App. D.4 repeats the multiplier analysis on SciQ and PIQA for bilingual μP and monolingual tuning, but not for the bilingual full HP sweep. We chose these benchmarks because monolingual models at 150M–1.43B score measurably above the random baseline on them, while pilot evaluations on MMLU and Arabic MMLU at the same scales remained at random across the full grid. Our data-multiplier analysis (§4.4) fits a log-linear curve through the monolingual points and inverts it at the bilingual target; this fit requires signal above random, which MMLU-class benchmarks did not deliver at our scales. The three benchmarks cover science knowledge, reading comprehension (open-book SciQ), and physical commonsense. Benchmarks that emphasize target-language syntax, generative fluency, or culturally specific knowledge may show different multiplier magnitudes, and our results do not speak to them. The natural follow-up is to scale the evaluation in benchmark difficulty by running stronger training recipes. Natively authored Arabic benchmarks such as Arabic MMLU, Belebele-AR, and AlGhafa are too hard for models in our regime: at 150M–1.43B parameters trained on 200M unique Arabic tokens they score at or near chance, so they cannot discriminate between our paradigms and cannot support a multiplier fit. The constraint is real, since it leaves us without a benchmark that is both tractable at our scale and free of translation artifacts. Our downstream claims are therefore about the relative ordering of pre-training strategies on a benchmark with measurable signal at these scales, not about absolute Arabic capability; confirming that ordering artifact-free requires a training recipe strong enough to produce signal on a natively authored benchmark.

Translated evaluation set. The Arabic versions of ARC Easy, SciQ, and PIQA are machine-translated from English with NLLB-200-3.3B (Team et al., 2022). Translated text retains English-aligned syntactic and statistical artifacts ("translationese"), and since bilingual models are trained on English, an artifact-driven component of the bilingual advantage cannot be ruled out from our data alone. A natively authored Arabic benchmark would still separate target-language capability from translation effects more cleanly than our current design; we flag this as the cleanest follow-up experiment.

Single language pair. We evaluate a single pair, Arabic–English. These two languages are typologically distant (different language families, scripts, and morphology), which makes our measurements

a relatively hard case for cross-lingual transfer; we expect closer pairs such as German–English or French–English to show effects at least as strong. Extending the evaluation to additional pairs is a natural follow-up. A second open question is the tokenizer: we use the XGLM tokenizer, which has strong subword coverage on Arabic, English, and a broad set of other languages. Its shared vocabulary favors Arabic–English transfer, so the multiplier magnitudes we report likely absorb a tokenizer-specific contribution. Studying other tokenizers, in particular on very low-resource languages where subword coverage is weaker, would isolate how much of the mixing advantage survives under less favorable tokenization.

Arabic as a proxy for scarcity. Arabic is a well-resourced language used here as a controlled stand-in for a data-constrained target (§3). Languages that are scarce in reality combine a small corpus with noisier text, narrower domain coverage, and weaker tokenizer support, and our multipliers should be read as a best case relative to that setting.

Static mixing ratios. We restrict to static mixing ratios; dynamic schedules that shift the ratio across phases are an orthogonal axis we leave to future work.

Single-seed runs. Each cell of our grid is trained once. We ran approximately 1000 models in total across the four scales; replicating the grid across multiple seeds would have multiplied total compute beyond our budget, and we do not report error bars across seeds. As a partial noise bound, the ANOVA in App. G reports residual variance at each scale: the two-way residual is 4–6% across 380M, 600M, and 1.43B, and the three-way residual stays below 7% at 380M and 600M. At 1.43B the three-way residual is larger but is driven by genuine interaction structure between $R_{\max}$, λ, and η rather than by seed noise (§G). Full seed replication is the cleaner measurement and is left to future work.

HP grid design. Our 5×5 HP grid was fixed across all model scales, and the per-scale optimum frequently lands on grid boundaries (App. F): $\eta = 0.01$ (the largest grid value) at smaller scales, and $\lambda = 0.01$ (the smallest grid value) at every scale. The true optimum may therefore lie outside our grid, and the HP sensitivity we measure at 1.43B is partly a property of the loss landscape and partly an artifact of the grid being off-center. A more informative analysis would have used a per-scale grid centered on the μP-predicted HP, with sufficient range in both directions to bracket the per-scale optimum. We recommend this design for follow-up work.

K Compute resources

We trained approximately 1000 models in total across the four scales. Each run was performed on a single GPU node, using either NVIDIA A100 or H100 GPUs. For the monolingual 20B-token runs, wall-clock times on A100 were 18.8 hours at 150M, 35.9 hours at 380M, 51.0 hours at 600M, and 95.0 hours at 1.43B. Bilingual runs varied in token budget across scales: 2.28B tokens in 2.7 hours (A100) at 150M, 12.17B tokens in 19.6 hours (A100) at 380M, 28.51B tokens in 62.0 hours (A100) at 600M, and 97.31B tokens in 191.0 hours (H100) at 1.43B.