WIDER BY HOW MUCH? LEARNING RATES SHAPE VISION-LANGUAGE ADAPTER COMPARISONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapter width in frozen-backbone vision-language models is often chosen from a width sweep trained at one shared learning rate. We show that the measured advantage of wider adapters depends strongly on that rate. Across five Perceiver-adapter widths and frozen language models from 1.5B to 14B parameters, the widest adapter beats the narrowest by 0.7–1.0 nats under a fixed rate at 1.5B–7B; selecting the rate at one width and sharing it removes 69–90% of that gap, and at 14B the narrowest cell varies by 0.57 nats across runs, so no ordering is determined there. Under the transferred rates the widest adapter is best or within 0.033 nats of best and 0.13–0.25 nats ahead of the narrowest. Changing the shared rate also changes Q-Former width comparisons, and the fixed-rate deficit extends to answer likelihood for the four narrower widths at 3B and all five at 14B, and to VQA accuracy after instruction tuning at 3B. A width-adjusted coordinate, the learning rate times the width, predicts held-out validation loss with 34–60% lower error than the rate alone below 14B. A one-anchor rule that rescales the reference rate by the square root of the adapter parameter-count ratio removes the per-width search, and at 14B its rates coincide with the sampled bracket minima at the two widest adapters. Matching the anchor's effective weight decay removes much of the rule's advantage at wide adapters, so the rule transfers the learning rate together with its decay. At 14B a single coordinate leaves width-dependent residuals; clipping settings barely change the narrowest adapter's loss at its rule rate, whereas halving the head count of the widest adapter at fixed parameter count changes its loss by -0.07 to $+0.12$ across two seeds. Connector comparisons should report learning-rate sensitivity or transfer the rate across widths.

1 INTRODUCTION

Modular vision-language models (VLMs) can connect a frozen vision encoder to a frozen large language model (LLM) through a trainable adapter (Alayrac et al., 2022; Li et al., 2023), or do so as a first alignment stage before the LLM is tuned (Liu et al., 2023). With both backbones fixed, the adapter width controls an important aspect of the trainable bridge's capacity. Connector studies compare architectures and sizes via ablation within a single training recipe (McKinzie et al., 2024; Tong et al., 2024; Laurençon et al., 2024), so the widths being compared share a single learning rate (LR). If the loss difference between two widths depends on which shared rate was used, such a sweep does not identify capacity, and a design conclusion drawn from it is conditional on the recipe. In this work, we show how large that dependence is and test a possible solution that needs no per-width search. Works on hyperparameter transfer show that the relationship between width and learning rate depends on the parameterization and optimizer (Everett et al., 2024), with μP giving the solution for the Adam case (Yang et al., 2021). In the frozen-backbone setting, we therefore ask how strongly LR assignment changes width comparisons and whether a reference-width rate can be transferred without a separate search at each width. To study this question, we train Perceiver-style adapters at five widths between a frozen vision encoder and frozen LLMs spanning an order of magnitude in size, under three protocols: a fixed shared rate, a rate selected at a reference width and then shared, and a width-aware rule that rescales the reference rate by adapter parameter count.

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.

Our findings and contributions can be summarized as follows:

- We show that evaluating all adapter widths at a single shared learning rate can substantially inflate the apparent advantage of wider adapters. When the rate is instead selected at a reference width and shared across widths, most of this gap disappears for 1.5B–7B parameter backbones, and changing the shared rate alone is enough to reorder Q-Former comparisons. The fixed-rate deficit also carries over to downstream answer likelihood, for widths XS–L at 3B and for all tested widths at 14B, and to VQA accuracy after instruction tuning at 3B (Section 4.1, Appendices S and T).
- A width-adjusted coordinate, ηd, predicts held-out validation loss substantially better than the learning rate alone for backbones below 14B, with the exponent bracketed rather than pinned (Section 4.2).
- Building on this coordinate, the one-anchor rule $\eta \propto N_A^{-1/2}$ requires no per-width search beyond the initial reference-rate selection. It corrects over-stepping at the widest adapter and matches the sampled bracket minima for the L and XL adapters at 14B, and a weight-decay control confirms that it transfers the learning rate jointly with its effective decay (Section 4.3, Appendix C).
- At 14B, a single coordinate leaves width-dependent residuals. Gradient clipping has little effect on the loss of the narrowest adapter at its rule rate, whereas altering the head configuration of the widest adapter at fixed width and parameter count changes its loss by amounts comparable to the XS–XL gap under the rule, with a sign that differs between seeds (Sections 4.3 and 5).

2 Related work

Scaling laws and multimodal scaling. Scaling laws describe how loss scales with model and dataset sizes (Kaplan et al., 2020; Hoffmann et al., 2022). Later work has extended them to transfer learning (Hernandez et al., 2021), data-constrained training (Muennighoff et al., 2023), vision transformers (Zhai et al., 2022), and contrastive image-language pretraining (Cherti et al., 2023), as well as to architectural choices such as mixture-of-experts (MoE) routing (Clark et al., 2022; Zhao et al., 2025). Multimodal scaling studies mostly vary whole systems (Shukor et al., 2025; Tian et al., 2025; Nezhurina et al., 2025) or visual token count at fixed adapter capacity (Li et al., 2025). Our comparison concerns the trainable connection rather than whole-system scaling. Table 1 contrasts the studied axes. The listed studies do not vary the adapter capacity or LR as scaling axes, and their trained components differ from a bridge between frozen backbones. We therefore compare the experimental designs and their scope rather than the numerical values of fitted laws.

Table 1: Axes varied as scaling variables by prior multimodal studies and this work. The prior studies do not treat adapter capacity or learning rate as an axis. Shukor et al. (2025) vary resolution, and thus token count, only in a side ablation.

Study	LLM size	Data	Visual tokens	Adapter capacity	Learning rate
Aghajanyan et al. (2023)	✓	✓	—	—	—
Li et al. (2025)	✓	—	✓	—	—
Shukor et al. (2025)	✓	✓	—	—	—
This work	✓	—	—	✓	✓ (3 protocols)

Hyperparameter transfer. Maximal update parameterization (μP) predicts that, when training with Adam (Kingma & Ba, 2014) as the optimizer, the learning rate of each hidden layer should scale inversely with its fan-in (the number of input units), so that the size of feature updates stays constant as width grows (Yang et al., 2021). Similarly, Everett et al. (2024) measures how the optimal LR exponent depends on parameterization and optimizer. Our rule $\eta \propto N_A^{-1/2}$ is motivated by their inverse-width scaling. Because N_A has terms linear in d, the rule is not exactly d^{-1} (at 14B the XS/M multiplier is 2.59 against $d_M/d_{XS} = 3$), and we apply it with the exact parameter counts. Recent work qualifies what makes such a transfer work. Kosson et al. (2026) find that weight decay, not the parameterization, stabilizes update magnitudes for most of the training, with μP acting mainly like a warmup. Kalra & Barkeshli (2026) attribute most of μP's advantage under AdamW (Loshchilov & Hutter, 2017) to the embedding-layer learning rate. Mlodozeniec et al. (2026) extend transfer to per-module rates, depth, batch size, and duration in dense LLMs. Chen et al. (2026) derive

learning-rate scaling across LoRA ranks on frozen pretrained weights, including vision-language settings, and Kalra & Barkeshli (2026) also ablates a frozen embedding layer. Our setting differs in what is trained. We only trained a newly initialized dense connector between two independently pretrained fixed backbones. We do not propose a new parametrization, and our rule omits both the initialization and the output scaling of full μP. Instead, we ask whether width comparisons in connector ablations depend on the learning-rate protocol; if they do, conclusions from single-recipe sweeps are confounded with the recipe itself. We quantify this dependence and test a solution that needs little hyperparameter search.

Optimization-aware scaling and adapters. Several scaling studies make optimization choices explicit, such as the LR schedule (Tissue et al., 2025), weight decay and batch size (Bergsma et al., 2025), and the learning rate and cooldown (McLeish et al., 2025). Modular VLM adapters include cross-attention (Alayrac et al., 2022), Q-Former (Li et al., 2023), linear projections (Liu et al., 2023) and Perceiver-style resamplers (Alayrac et al., 2022; Jaegle et al., 2021). Prior studies compare connector architectures within their reported training recipes (McKinzie et al., 2024; Tong et al., 2024; Laurençon et al., 2024). Our results motivate an explicit LR-sensitivity check in such comparisons.

3 METHOD

3.1 SETTING AND NOTATION

A frozen SigLIP-SO400M/14-384 encoder (Zhai et al., 2023) emits 729 patch tokens. A trainable Perceiver Resampler (Alayrac et al., 2022; Jaegle et al., 2021) of width d maps them to T visual tokens prepended to the text embeddings of a frozen Qwen2.5 LLM (Qwen Team, 2024). The resampler has two blocks, SwiGLU feed-forward layers (4$\times$ expansion) and LayerNorm, with 4, 8, 8, 8, and 16 heads across the five widths. Head dimension is therefore not constant, and it matters (a 14B head-count control moves XL by -0.07 to $+0.12$ nats across two seeds; Section 5). Only the adapter is trained, with next-token loss on text tokens. We evaluate various widths XS/S/M/L/XL which are $d \in \{256, 512, 768, 1024, 1536\}$ with N_A from 3.3M to 104M trainable parameters (Appendix A). Unless stated otherwise, T=64 and training is one pass over the 552,546 LLaVA-Pretrain pairs, that is, 17,267 optimizer steps at an effective batch size of 32, with 345 warm-up steps and cosine decay to zero. We use AdamW ($\beta = (0.9, 0.95)$, weight decay 0.1), bfloat16 and gradient clipping at 1.0. Since weight decay acts as $1 - \eta\lambda$ with λ fixed, changing η also changes the effective decay, so the protocols compare learning-rate assignments as a whole. Validation loss is the token-mean caption NLL evaluated every 500 optimizer steps on a fixed held-out file of 5,582 image–caption pairs (the same file for every run). We report the minimum across periodic evaluations. An additional evaluation at the end of training is recorded separately and does not update this minimum. Where the two differ materially, we report both. Table 3 and Figure 1 report cell means over the seed-42 runs of the width-sweep groups (Table 19; three M cells mix micro-batch decompositions, Table 7); same-configuration runs of other groups enter the pooled fit of Section 4.2 but not the cells. Some fixed-versus-selected comparisons at 7B and 14B also differ in micro-batch decomposition. The 14B bracket and rule-versus-shared comparisons and the same-seed repeats of Table 7 are decomposition-matched except for its three M cells (Appendix J, which also lists wall times and hardware).

3.2 THREE LEARNING-RATE PROTOCOLS

We compare adapter widths under three protocols. Fixed: one LR, 10^{-4}, for every width and scale, the conventional design. Scale-level selected: for each LLM, a four-point LR sweep at the mid width M (three points reported at 7B) picks the best rate, 3×10^{-4} at every fitted scale (Table 2), which is then shared across widths. Width-aware: the selected rate at M is transferred to every other width by

$$\eta(w) = \eta_M \sqrt{N_A^{(M)} / N_A^{(w)}}, \tag{1}$$

so that $\eta\sqrt{N_A}$ is held fixed. Since the quadratic-in-d terms dominate N_A at these widths, this is approximately ηd fixed. At 14B this gives $7.8, 4.3, 3.0, 2.3, 1.6 \times 10^{-4}$ for XS–XL. After the learning rate is selected at one reference width, the rule requires no further per-width search. We evaluate it with matched controls (the same M run serves both the selected and the width-aware protocol), per-width LR brackets, seed replicates and held-out widths whose LR and loss were recorded before the runs.

3.3 WIDTH-ADJUSTED LEARNING-RATE ANALYSIS

Motivated by inverse-width learning-rate scaling, which is the standard parameterization prediction behind μP, we test whether validation loss at the fixed budget is approximately described by a width-adjusted coordinate $x = \eta\, d^{\alpha}$ with $\alpha \approx 1$. This is an empirical hypothesis rather than a consequence of matched update magnitudes, and x is a learning-rate coordinate rather than a measured Adam update. For each LLM we pool every run in the primary analysis set at T=64, D=552K over all widths and all learning rates we have (5×10^{-5} to 8×10^{-4}; this primary set was fixed before the W, X and Y series of Table 19, which enter no fit), fit a quadratic in $\log x$, and report the residual RMSE as a function of α. We test whether validation losses from different adapter widths and learning rates approximately align along a common trend when plotted against x. We assess this alignment with two quantities: the error of the fit in predicting the loss of held-out widths and learning rates (Section 4.2), and the residual differences between widths at matched values of x. The primary loss summaries and fits condition on runs without an unrecovered training spike. The two runs that did not train are kept in the record and analyzed separately (Appendix H).

Seed spread at width M ranges from 0.026–0.027 (3B, 7B) to 0.15–0.17 (14B and 1.5B, one outlier seed each), six 14B XS runs at the shared 3×10^{-4} span 0.57, two runs with identical configuration and seed on the same host differ by 0.085–0.25 at 7B, and runs of one configuration on different hosts differ by up to 0.17 at 3B (Appendix B). We therefore interpret differences relative to the replicate spread of the cells involved (Table 7) and draw no firm ranking from 7B and 14B differences below about 0.09, the same-seed replicate spread there.

4 EXPERIMENTS

4.1 A SHARED LEARNING RATE INFLATES THE ADVANTAGE OF WIDE ADAPTERS

The conventional rate under-trains the adapter at every scale. In the scale-level sweep at the reference width M (Table 2), 3×10^{-4} is the best tested rate for every LLM and improves on 10^{-4} by 0.43–0.59 nats, and rates above 3×10^{-4} give equal or higher loss at every scale (Appendix E).

Table 2: Scale-level learning-rate sweep: best validation loss at T=64, width M (d=768); Δ is the change from 10^{-4} to 3×10^{-4}. The 7B cell at 5×10^{-5} is omitted because its single run failed our run-record verification (see unverified runs, ^{u} in Table 19).

LLM	5×10^{-5}	10^{-4}	2×10^{-4}	3×10^{-4}	Δ
1.5B	3.651	3.239	2.805	**2.713**	-0.53
3B	3.606	3.086	2.671	**2.660**	-0.43
7B	—	3.079	2.618	**2.492**	-0.59
14B	3.406	3.115	2.859	**2.522**	-0.59

Learning-rate assignment changes the apparent benefit of width, not only the overall level of the loss (see Table 3 and Figure 1). Under the fixed 10^{-4}, XL is the best width at all four scales and leads XS by 0.7–1.0 nats at 1.5B–7B. Under the selected 3×10^{-4}, most of that lead disappears. The XS–XL gap shrinks to 0.30, 0.08 and 0.17 nats at 1.5B, 3B and 7B, a reduction of 69–90%. XS gains 0.5–0.7 nats from the higher rate, whereas XL changes by less than 0.06 at 1.5B–7B, comparable to the replicate spread of Table 7. Most of the fixed-rate gap therefore disappears under the higher shared learning rate, so a fixed-rate sweep does not isolate the effect of capacity.

What remains once the rate is selected is not a resolvable ranking. At 1.5B–7B, S through XL lie within 0.07–0.13 of one another, comparable to the spread between runs of a single configuration (Table 7), and the best cell moves away from XL at every scale by 0.007–0.12. At 14B the XS cell itself is unstable: six runs over five seeds at 3×10^{-4} fall into two groups, 2.49–2.59 and 3.06 (Appendix B), so no XS–XL ordering is determined there. Micro-batch decomposition differs between the fixed and selected rows at 7B and 14B; a decomposition-matched rerun reproduces eight of the ten fixed-rate cells within 0.012 (Appendix J).

Other adapter families. We found that the reordering is not specific to the Perceiver Resampler. A Q-Former width sweep at 3B and 14B under the same two shared settings (Table 4) shows the same

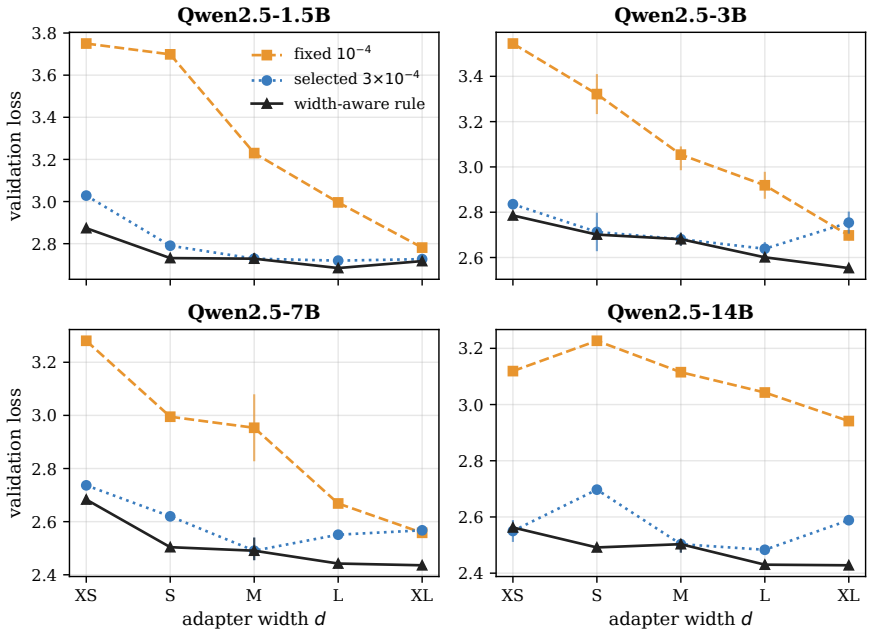


Figure 1: Validation loss against adapter width under the three protocols of Section 3.2 (T=64). Points are the cell values of Table 3. The vertical bars show the range of the runs in cells with more than one run. Under the fixed rate, the widest adapter wins at every scale. Under the width-aware rule, the curves are predominantly decreasing or saturating.

Table 3: Best validation loss by adapter width under the three protocols (T=64). Bold marks the lowest value per row. A cell with more than one run shows their mean; Table 7 lists those runs and their ranges.

LLM	Protocol	XS	S	M	L	XL
1.5B	fixed 10^{-4}	3.750	3.699	3.230	2.996	**2.781**
	selected 3×10^{-4}	3.028	2.790	2.729	**2.720**	2.727
	width-aware	2.874	2.732	2.729	**2.684**	2.717
3B	fixed 10^{-4}	3.545	3.322	3.054	2.919	**2.698**
	selected 3×10^{-4}	2.835	2.713	2.681	**2.639**	2.754
	width-aware	2.786	2.701	2.681	2.601	**2.553**
7B	fixed 10^{-4}	3.281	2.995	2.953	2.668	**2.558**
	selected 3×10^{-4}	2.737	2.620	**2.491**	2.551	2.567
	width-aware	2.683	2.504	2.491	2.442	**2.436**
14B	fixed 10^{-4}	3.119	3.227	3.115	3.043	**2.941**
	selected 3×10^{-4}	2.550	2.697	2.503	**2.483**	2.588
	width-aware	2.563	2.492	2.503	2.430	**2.428**

pattern. At 3B the shared 10^{-4} gives a monotone curve with XL best, while 3×10^{-4} reorders it and compresses the spread from 0.764 to 0.125. At 14B both rates are non-monotone with M or M/L best. A fixed-capacity MLP projector trails both adapter families at every token count (more details on this experiment are in Appendix F).

4.2 Width adjustment improves held-out loss prediction

Replacing the learning rate by the width-adjusted coordinate makes validation loss more predictable at conditions the fit has not seen (Figure 2, Table 5). With α fixed at 1, the fit predicts the loss of a

Table 4: Q-Former capacity sweep at T=64 (best validation loss of 20 runs). Bold marks the best width per column.

Width	3B, 10^{-4}	3B, 3×10^{-4}	14B, 10^{-4}	14B, 3×10^{-4}
XS	3.517	2.760	3.084	2.496
S	3.159	2.648	2.884	2.390
M	2.967	2.655	**2.696**	**2.376**
L	2.893	**2.635**	3.029	**2.376**
XL	**2.753**	2.670	2.929	2.537

Table 5: Held-out prediction RMSE of the $\eta\, d^{\alpha}$ fit, averaged over splits; points = distinct (width, LR) cells after averaging the runs of each cell over all seeds α_{fit} is chosen by in-sample residual on the fitting part of each split.

		leave-one-width-out			leave-one-LR-out		
LLM	points	α=0	α=1	α_{fit}	α=0	α=1	α_{fit}
1.5B	16	0.289	**0.145**	0.184	0.185	**0.096**	0.121
3B	24	0.227	0.110	**0.097**	0.167	0.110	**0.076**
7B	16	0.180	**0.093**	0.096	0.214	**0.086**	0.108
14B	30	0.219	0.183	**0.168**	0.181	0.163	**0.162**

held-out width or learning rate with an error of 0.086–0.145 nats at 1.5B–7B, against 0.167–0.289 for η alone, a reduction of 34–60%. At 14B the reduction is only 10–17% (0.16–0.18 against 0.18–0.22). The rule's exact coordinate $\eta\sqrt{N_A}$ gives the same picture with slightly larger reductions (Table 15).

The exponent is bracketed rather than pinned: held-out error is lowest at $\alpha = 1$ at 1.5B and 7B, the in-sample residual over a grid of exponents is minimized at $\alpha = 0.5$–0.75 (Appendix H, Table 16), and selecting the exponent on the fitting data brings no consistent gain. We therefore treat α as lying between about 0.5 and 1 and use $\alpha = 1$ as the prespecified predictor.

In this coordinate the fixed-rate width sweep reads as a slice through one trend rather than as a capacity curve: at 10^{-4}, XS sits at $\eta d = 0.026$, deep in the under-trained region, while XL sits at 0.154. At matched coordinate the 3B widths nearly coincide: configuration-mean losses at $\eta d \geq 0.08$ differ across widths by at most 0.04 nats, and in the run-level fit the mean residual of the XL runs is 0.03–0.05 nats below that of S, M and L (0.04–0.11 at the in-sample-best exponent), an order of magnitude smaller than the fixed-rate spread. This residual is consistent with a capacity benefit of the widest adapter, but head count and head dimension change with width (Section 3), so it does not isolate capacity.

The pooled trend is predominantly decreasing over the tested range (the 3B quadratic turns up by 0.02 nats at its right end, within replicate spread), so these experiments do not establish an upper stability bound on the rate at 7B and below (Section 4.3). Individual widths nevertheless over-step: at the shared 3×10^{-4}, XL is 0.13–0.20 nats worse than at its rule rate at 3B–14B. The next subsection tests whether the rate selected at M transfers to the other widths.

4.3 Width-aware learning-rate transfer

Transferring the reference rate with Eq. 1 gives width curves that are predominantly decreasing or saturating, with XL the best width or within 0.033 of the lowest loss at every scale (Table 3). The rule maintains the higher rates required by the narrow widths and lowers the rate of the widest adapter, thereby removing the over-stepping observed in Section 4.2 (0.13–0.20 nats at XL). At 14B, rule-rate seed replicates differ by 0.011 for XS and 0.016 for XL, against 0.57 over the six shared-rate XS runs (Appendix B). Table 3 averages the runs of each cell, so its 14B M value (2.503) differs from the single anchor run (2.515) of the matched 14B suite.

LR brackets. Per-width brackets at 14B test whether the transferred rates are near the best rate of each width (see Figure 3, more details in Appendix E). For the L and XL widths, the sampled minimum lies at the rule rate and is bracketed on both sides. At XL, the two alternatives, the shared

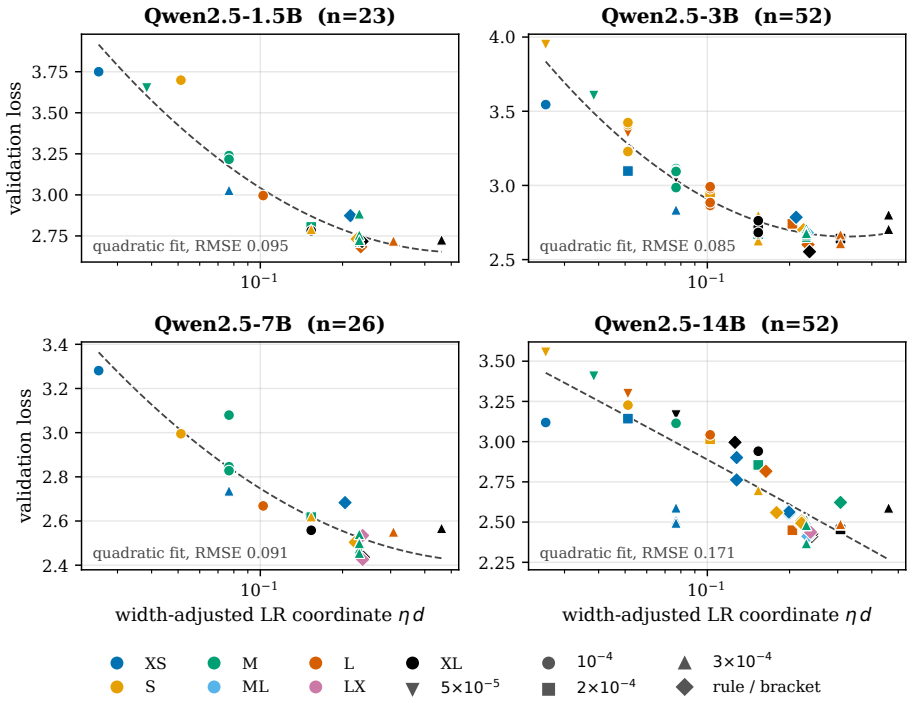


Figure 2: Validation loss versus the width-adjusted learning-rate (LR) coordinate ηd, where η is the learning rate and d is the adapter width. Each point is one run from the primary analysis set (T=64, D=552K). Color indicates width, marker shape indicates learning rate, and lines show quadratic fits in $\log(\eta d)$. For backbones up to 7B parameters, runs follow the fitted trend closely, so ηd largely captures the effect of width. At 14B, runs with the same ηd no longer overlap. They are separated by width, and the direction of this separation is not consistent across widths. ML and LX mark the held-out widths d=896 and d=1280 (Section 4.3).

rate (exponent 0) and a linear rescaling by the parameter-count ratio (exponent 1), both give higher loss. For XS, the shared and transferred rates give similar losses, 2.589 and 2.563, and for S, the outcome varies between runs, as described below. These brackets support useful transfer. They do not establish the optimal rate at every width, and other seeds reach lower losses at some widths (Appendix B).

Prospective held-out widths. We recorded the rule's rate and a predicted loss for a new width d=896 at 3B and 14B before those runs and, after refitting with them, for d=1280 at 7B and 14B, the predictions came from a descriptive fit of loss against N_A with one intercept per LLM (Appendix H). At d=896, the observed losses were 2.412 at 14B and 2.681 at 3B against predictions of 2.454 and 2.629. At d=1280, the error was 0.014 at 14B and 0.085 and 0.023 over two seeds at 7B. Three of the four errors are within 0.053. The 7B cell has a mean error of 0.054 across two seeds, with losses that differ by 0.108.

Effective weight decay. Part of what the rule transfers is regularization. With λ fixed, the rule approximately halves the per-step decay $\eta\lambda$ at XL relative to the anchor and raises it 2.75× at 3B XS. Rerunning eight-rule cells (twelve runs) with λ rescaled so that $\eta\lambda$ equals the anchor's (Appendix C) raises L at 3B and 14B and XL at all three scales by 0.06–0.14 nats and lowers 3B XS and S by 0.04–0.06. Much of the rule's gain at wide adapters, therefore, comes from the lower decay. With $\eta\lambda$ held at the anchor's value, the rule beats the shared 3×10^{-4} at 14B XL by 0.02–0.05 rather than 0.16, and at L not at all (0.005–0.033 worse). The rule, as evaluated, transfers the pair $(\eta, \eta\lambda)$ and remains useful in that form.

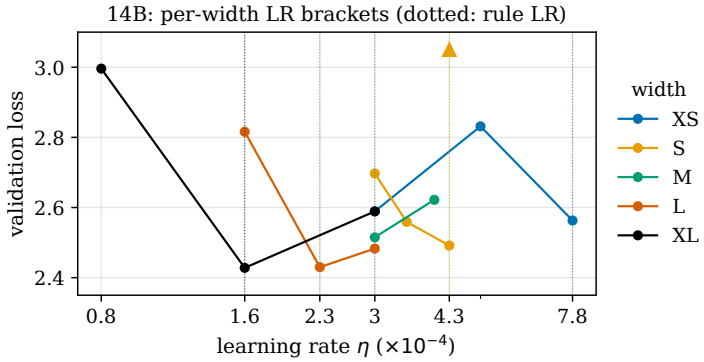


Figure 3: Per-width learning-rate brackets at 14B (T=64). Each curve is one width. The dotted verticals mark the rate the rule of Eq. 1 prescribes. For L and XL, the sampled minimum coincides with the rule rate and is bracketed on both sides. The narrow widths are not monotone in η: at its rule rate S showed an unrecovered spike in one of six same-configuration runs (The triangle marks this off-scale loss. The S point averages the other five), and XS is worse at 5×10^{-4} than at either neighbor in both of its runs.

Run-to-run variability at the prescribed rate. The transferred rate is not uniformly safe at the narrow 14B widths. The rule sends 14B S model variant to 4.3×10^{-4}; of six runs of this configuration, one spiked at steps 1,150–1,700 and did not recover (best 3.89), while five, on another host, reached 2.479–2.507 (Appendix B). The S cell of Table 3 averages the three seed-42 runs among those five. At 5×10^{-4}, XS reached 2.901 and 2.762 in two runs, worse than at either neighboring rate, so the XS bracket is not monotone. One failure in six, with host confounded with outcome, does not separate a rare failure from a host effect; no 1.5B–7B run shows such a spike.

At 14B, one coordinate is not enough. At 14B, the width-adjusted coordinate predicts less and leaves more unexplained. The best tested exponent leaves an in-sample residual of 0.14 nats against 0.07–0.09 at 1.5B–7B (Table 16), and held-out errors are 0.16–0.18 against 0.09–0.15, with ηd reducing error by only 10–17% (Table 5). Runs at matched coordinate show where the variation sits: at $\eta d \approx 0.077$, XS, M, and XL differ by 0.62 nats, or 0.37 with the six-run XS mean; at $\eta d \approx 0.154$, S and XL differ by 0.24; and at $\eta d \approx 0.31$ the ordering reverses, with M trailing XL by 0.17.

Spikes are also confined to 14B. Among the 132 1.5B–14B runs with training logs (50-step resolution, Appendix I; the detector cannot assess the warm-up itself), post-warm-up spikes occur in exactly four runs, all at 14B: S at its rule rate (unrecovered), XL at 1.6×10^{-4} and 3×10^{-4} and the held-out width at 2.6×10^{-4} (all recovered). No 1.5B, 3B or 7B run spikes, including every XL run at 10^{-4} and at the rule LR. The residual is concentrated at the narrow end. XS departs most from the pooled trend, in loss and in run-to-run outcome, while L and XL follow the rule. Two controls narrow the candidates without identifying a cause (Section 5). At the XS rule rate, clipping settings change its loss by less than 0.015, whereas halving the head count of XL at its rule rate changes its loss by $+0.12$ at one seed and -0.07 at another.

5 Discussion

Our central observation is that a width sweep run at a single shared learning rate does not isolate width. It measures width and learning-rate assignment together, and at architecture sizes from 1.5B to 7B the second factor dominates. Raising the shared rate improves the narrowest adapter by 0.5–0.7 nats while moving the widest by less than 0.06 nats. Once the width is folded into the learning-rate coordinate, most of the apparent width effect disappears. What remains is a small advantage for the widest adapter at 3B (Section 4.2), and even this cannot be read as a pure capacity effect, since head count and head dimension change along with width.

The practical consequence is that a connector comparison is only as meaningful as the learning-rate protocol behind it. We therefore treat the chosen rate and how sensitive the ordering is to that rate as part of the result rather than as an implementation detail. Eq. 1 keeps this affordable, because once the reference rate is selected at one anchor, the rule supplies the rate for every other width without further search. One caveat matters for how the gains should be read. In the form we evaluate, the weight-decay coefficient is held fixed, so transferring the learning rate also transfers the effective decay, and the control in Appendix C shows that much of the wide-adapter gain arrives through this lower effective decay. Whether a full maximal-update parameterization would eliminate the remaining residual at 14B remains an open question.

At 14B, the rule proves to be useful without being complete. The sampled brackets place the L and XL minima at the rule rate, so transfer works where it matters most for selection. Yet a single coordinate still leaves width-dependent residuals, and the narrowest adapter departs the most, splitting into two outcome groups at the shared rate (Section 4.3). We tried to pin down the cause with the controls in Table 11 (Appendix E), but none of them resolve it. The XL head-count effect reverses sign between the two seeds we tested. At XS, the head and effective-decay changes we tested move the loss by less than 0.1 nats, clipping barely matters at the rule rate, and turning clipping off leaves both high-loss shared-rate runs near 3.07. At S, the effective decay control changes the loss by less than 0.03. We therefore report the 14B residual as an open feature of this regime rather than attribute it to any single factor.

As this work's limitations, our conclusions come from a narrow setting. Our primary experiments pair two-block Perceiver adapters with one vision encoder (SigLIP-SO400M) and one LLM family (Qwen2.5), trained on a single dataset at a fixed exposure with $T{=}64$, an effective batch size of 32, and a fixed weight-decay coefficient. Within this setting, we bracket the exponent of the width coordinate between about 0.5 and 1 rather than estimate it, and depth also moves the loss (Appendix P). Beyond Perceivers, the Q-Former sweep shows the same learning-rate sensitivity (Table 4), but neither it nor the MLP check (Appendix F) is enough to establish the transfer rule across adapter families.

Finally, our primary metric is pretraining validation loss, and the extent to which its conclusions carry downstream depends on scale. Downstream answer NLL registers the large fixed-rate deficit for XS–L at 3B and for all widths at 14B, is task-dependent at 7B, and does not preserve fine width rankings (Appendix S). After instruction tuning with one recipe at 3B, the fixed-rate deficit at XS–L survives as 5–20 accuracy points, while the two higher-rate rows stay within 4 VQAv2 points of each other (Appendix T). We let other scales and recipes open for future research.

6 Conclusion

In this work, we studied how much of an adapter-width comparison is really about width. Between different-sized frozen vision and language backbones, the answer depends heavily on how the learning rate is assigned. When the rate is selected at a reference width and held constant across the sweep, most of the gap that initially favored the widest adapter across sizes from 1.5B to 7B disappears. The same change reorders Q-Former sweeps, and the deficit left by a fixed rate carries through to downstream answer likelihood. As shown through a width-adjusted learning-rate coordinate experiment, the fixed-rate width curve becomes a slice through a single trend, and this coordinate predicts held-out validation loss substantially better than the rate alone below 14B. A one-anchor rule that rescales the reference rate by adapter parameter count turns this observation into a usable procedure. It removes the need for a per-width search, stops the widest adapter from overstepping, and matches the sampled bracket minima for the L and XL adapters at 14B, while carrying regularization along with the learning rate.

For connector design, our finding is that adapter width and its training protocol should be evaluated together. A width comparison is most informative when it reports the rate at which it was run and how sensitive its ordering is to that rate, or when it transfers the rate across widths using a rule like the width-aware rule we tested here, while keeping the decay treatment fixed. We acknowledge that the width-aware rule we evaluated in this work is a starting point rather than a finished account, since it leaves width-dependent residuals at 14B, concentrated in the narrowest adapter, the cause of which we could not isolate. Explaining these residuals, testing the rule on other backbone families and adapter types, and extending the instruction-tuning evaluation beyond 3B are open for future research.

AI USE STATEMENT

We used large language model assistants to provide feedback on hypotheses and experimental design, and to revise the writing to improve the use of scientific language. Additionally, LLM assistants are used to implement and debug code, and to run and record experiments. The authors selected the experiments, analyses, and interpretations, reviewed all retained code and text, executed the analysis code and checked its outputs against the run records, and verified bibliographic details against the original sources. The authors take responsibility for the final text, claims, code, and other artifacts.

ETHICS STATEMENT

This work trains only adapters between public models on a public dataset and introduces no new generative capability. Appendix R discusses broader impacts.

REPRODUCIBILITY STATEMENT

Appendices A–T follow the references in this PDF. The supplementary archive contains the training code, the resolved configuration of every run listed in its run lists (launch files where available), the per-run ledger of results (which flags superseded and unverified runs; the latter appear only in the exploratory appendix summaries marked ^{u} in Table 19), the run lists behind the figures and the width, held-out, control and downstream tables, the instruction-tuning configurations and evaluation outputs of Appendix T, and a script that regenerates Tables 3, 5, 15 and 16 and the data of Figures 1 and 2 from that list. Adapter checkpoints will be released on acceptance. Appendix J lists hardware and wall times.

REFERENCES

Armen Aghajanyan, Lili Yu, Alexis Conneau, Wei-Ning Hsu, Karen Hambardzumyan, Susan Zhang, Stephen Roller, Naman Goyal, Omer Levy, and Luke Zettlemoyer. Scaling laws for generative mixed-modal language models. In *ICML*, 2023.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

Shane Bergsma, Nolan Dey, Gurpreet Gosal, Gavia Gray, Daria Soboleva, and Joel Hestness. Power lines: Scaling laws for weight decay and batch size in LLM pre-training. In *NeurIPS*, 2025.

Nan Chen, Soledad Villar, and Soufiane Hayou. Learning rate scaling across LoRA ranks and transfer to full finetuning. *arXiv preprint arXiv:2602.06204*, 2026.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *CVPR*, 2023.

Aidan Clark, Diego de Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, et al. Unified scaling laws for routed language models. In *ICML*, pp. 4057–4086. PMLR, 2022.

Katie Everett, Lechao Xiao, Mitchell Wortsman, Alexander A. Alemi, Roman Novak, Peter J. Liu, Izzeddin Gur, Jascha Sohl-Dickstein, Leslie Pack Kaelbling, Jaehoon Lee, and Jeffrey Pennington. Scaling exponents across parameterizations and optimizers. In *ICML*, 2024.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017.

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. In *NeurIPS*, 2022.

Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention. In *ICML*, 2021.

Dayal Singh Kalra and Maissam Barkeshli. Quantifying hyperparameter transfer and the importance of embedding layer learning rate. *arXiv preprint arXiv:2605.21486*, 2026.

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

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014. URL https://arxiv.org/abs/1412.6980.

Atli Kosson, Jeremy Welborn, Yang Liu, Martin Jaggi, and Xi Chen. Weight decay may matter more than muP for learning rate transfer in practice. In *ICLR*, 2026. arXiv:2510.19093.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? In *NeurIPS*, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023.

Kevin Y. Li, Sachin Goyal, Joao D. Semedo, and J. Zico Kolter. Inference optimal VLMs need fewer visual tokens and more parameters. In *ICLR*, 2025.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101, 2017. URL https://arxiv.org/abs/1711.05101.

Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Anton Belyi, Haotian Zhang, Karanjeet Singh, Doug Kang, Hongyu Hè, Max Schwarzer, Tom Gunter, Xiang Kong, Aonan Zhang, Jianyu Wang, Chong Wang, Nan Du, Tao Lei, Sam Wiseman, Mark Lee, Zirui Wang, Ruoming Pang, Peter Grasch, Alexander Toshev, and Yinfei Yang. MM1: Methods, analysis & insights from multimodal LLM pre-training. In *ECCV*, 2024.

Sean McLeish, John Kirchenbauer, David Yu Miller, Siddharth Singh, Abhinav Bhatele, Micah Goldblum, Ashwinee Panda, and Tom Goldstein. Gemstones: A model suite for multi-faceted scaling laws. In *NeurIPS*, 2025.

Bruno Mlodozeniec, Pierre Ablin, Louis Béthune, Dan Busbridge, Michal Klein, Jason Ramapuram, and Marco Cuturi. Completed hyperparameter transfer across modules, width, depth, batch and duration. In *ICLR*, 2026. arXiv:2512.22382.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36:50358–50376, 2023.

Marianna Nezhurina, Tomer Porian, Giovanni Pucceti, Tommie Kerssies, Romain Beaumont, Mehdi Cherti, and Jenia Jitsev. Scaling laws for robust comparison of open foundation language-vision models and datasets. *arXiv preprint arXiv:2506.04598*, 2025.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Mustafa Shukor, Enrico Fini, Victor Guilherme Turrisi da Costa, Matthieu Cord, Joshua Susskind, and Alaaeldin El-Nouby. Scaling laws for native multimodal models. In *ICCV*, 2025.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *CVPR*, 2019.

Changyao Tian, Hao Li, Gen Luo, Xizhou Zhu, Weijie Su, Hanming Deng, Jinguo Zhu, Jie Shao, Ziran Zhu, Yunpeng Liu, et al. Navil: Rethinking scaling properties of native multimodal large language models under data constraints. In *NeurIPS*, 2025.

Howe Tissue, Venus Wang, and Lu Wang. Scaling law with learning rate annealing. In *NeurIPS*, 2025.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *NeurIPS*, 2024.

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. In *NeurIPS*, 2021.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR*, pp. 12104–12113, 2022.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

Guoliang Zhao, Yuhan Fu, Shuaipeng Li, Xingwu Sun, Ruobing Xie, An Wang, Weidong Han, Zhen Yang, Weixuan Sun, Yudong Zhang, Cheng-zhong Xu, Di Wang, and Jie Jiang. Towards a comprehensive scaling law of mixture-of-experts. *arXiv preprint arXiv:2509.23678*, 2025.

APPENDIX

The appendix reports the material that the main text relies on but does not need in full: parameter counts (A), seed noise (B), the effective-weight-decay control (C), training curves (D), the width-specific LR diagnostic and 14B brackets (E), other adapter families (F), the token sweep (G), held-out validation of the width-adjusted fit (H), the training-log spike scan (I), compute, inference cost and licenses (J–L), the experiment inventory (M), the token-by-width interaction (N), input resolution (O), adapter depth (P), boundary cases (Q), broader impacts (R), downstream NLL (S) and task accuracy after instruction tuning (T).

A ADAPTER PARAMETER COUNTS AND HEAD CONFIGURATION

Trainable parameters $N_A(d, N_L)$ in millions, from counted parameters of the trained modules; the output projection $d \times d_{\text{LLM}}$ makes the count weakly LLM-dependent. Heads are 4 (XS), 8 (S, M, L) and 16 (XL), so head dimension is 64, 64, 96, 128 and 96 respectively. Attention logits are scaled by the inverse square root of the head dimension, so head configuration is a structural difference between widths that the coordinate ηd does not model. The Q-Former adapters of Table 4 have 2.9/10.1/21.4/36.9/80.6M (3B) and 3.7/11.6/23.8/40.1/85.3M (14B) trainable parameters for XS–XL.

Level	d	1.5B	3B	7B	14B
XS	256	3.33	3.46	3.86	4.25
S	512	11.9	12.2	13.0	13.7
M	768	25.7	26.1	27.3	28.5
L	1024	44.8	45.3	46.9	48.5
XL	1536	98.6	99.4	102	104

Table 6: Trainable adapter parameters N_A (millions) by width and LLM.

B SEED NOISE AND REPLICATE SPREAD

Replicates of one configuration. Several Table 3 cells have more than one run of the same configuration and seed (a second run on another host, or the D=552K run named by sample count and by epoch). Table 7 lists them; the table cells show their mean. The spread is host-dependent: pairs on the RTX 5090 host agree to 0.002 (14B M/L/XL at 10^{-4}), whereas pairs that involve the A6000 hosts differ by 0.05–0.25 (3B S 0.17–0.18, one pair on one host and one pair involving a run of unrecorded host; 7B M at 10^{-4} 0.25 for two runs on the same host with the same seed). These are larger than the single-host seed spread at 3B below, so we draw no firm ranking from differences below about 0.1 at 3B, or below the listed replicate spread of a cell. In the one pair we tested directly, the variation was traced to non-deterministic GPU kernels: two 14B runs with identical configuration and seed are bit-identical for 50 steps when deterministic kernels are forced (fixed reduction order, math attention; about 25% slower) and separate from the third step with the default kernels; training then amplifies the difference: two 2,000-step 14B XS runs with identical configuration and seed differed by 0.39 nats at step 2,000, and two full 14B XL runs under the rule that differed by 0.53 at step 2,000 ended 0.016 apart. The deterministic comparison covers logged loss values over 50 steps and does not establish a single cause for all run-to-run variation observed here. A direct probe of sensitivity uses the deterministic setting with a deliberate kick: adding Gaussian noise of 5×10^{-3} times each tensor's RMS to the adapter after step 3 changed the step-2,000 validation loss of 2,000-step runs at the shared 3×10^{-4} by 0.021 and 0.382 for 14B XS (two noise seeds), by at most 0.012 for 14B M and XL, and by 0.03–0.15 for the three 3B widths; the two perturbation directions thus produced markedly different responses for the 14B narrow adapter, the 14B wide adapters were the least sensitive cells, and 3B trajectories carried a moderate sensitivity at every width. In the prespecified comparison of the 14B XS pair, the frozen-LLM response K_t did not lead the loss excursion by the required three logged points: its increase and the loss jump appeared within the same 10-step logging interval, so K_t is a same-time marker of the excursion, not a validated early-warning signal at this resolution.

Seed replicates. Seed replicates at width M and 3×10^{-4} (best validation loss; the seed-42 value is the anchor run and the seed-123/456 runs use 32×1 at 1.5B/3B and 16×2 at 7B/14B, so only the 3B triple is decomposition-matched): 3B seeds 42/123/456 give 2.695/2.668/2.678 (spread 0.027); 7B 2.476/2.501/2.499 (0.026); 1.5B 2.712/2.885/2.724 (0.17, one outlier seed); 14B 2.515/2.481/2.367 (0.15). At 14B under the shared 3×10^{-4} the XS cell is unstable: six runs over five seeds (seed 42 twice; 123, 456, 789, 1011) fall into two groups, 2.511, 2.589, 2.495 and 3.057, 3.065, 3.060 (range 0.57); the three high runs descend monotonically to their plateau with no spike, while L gives 2.483/2.487 (0.004); under the width-aware rule XS gives 2.563/2.552 (0.011) and XL 2.428/2.412 (0.016); with 8 heads instead of 16, XL gives 2.549/2.339 (0.21). The 7B held-out d=1280 cell spans 0.108 across two seeds. At the rule LR the spread is larger for the narrow widths: 3B XS 2.697–2.803 over seeds 42/123/456 (0.106) against XL 2.549–2.578 (0.030), and 7B XS 2.547–2.683 (0.136) against XL 2.436–2.471 (0.036); Table 3 reports the seed-42 runs, and with three-seed means XS remains the worst rule width at both scales while 7B L and XL become tied within 0.02 (2.442 against 2.454). Run-to-run nondeterminism alone is of the same size at 7B: two reruns of 7B M at 3×10^{-4} with identical configuration and seed on the same host (differing only in run name) reach 2.540 and 2.455, and their training losses already differ by 0.075 at step 50, so bf16 accumulation order and kernel selection are amplified by training rather than averaged out. Two further full runs of the same configuration with deterministic kernels are identical at every logged step and reach the same loss (2.5713), so the divergence under default kernels is attributable to this nondeterminism. We therefore draw no firm ranking from 7B and 14B differences below 0.09 and report replicates wherever such an ordering matters.

C EFFECTIVE-WEIGHT-DECAY CONTROL

Because λ is fixed, the rule changes the per-step decay $\eta\lambda$ together with the learning rate: relative to the anchor, XS receives 2.6–2.75×, S 1.5×, L 0.76× and XL 0.51–0.52× the anchor's $\eta\lambda$. To separate the two we reran eight rule cells (twelve runs) with λ rescaled per run so that $\eta\lambda$ equals the anchor's ($\lambda = 0.1 \cdot 3 \times 10^{-4}/\eta$), with everything else unchanged: the four non-anchor 3B widths for seed 42 (XS and XL also for seed 123), with M reused as the unchanged anchor, 7B XS and XL, and

Table 7: Runs of the same configuration (T=64, D=552K, seed 42) that feed one cell of Table 3: run group (with a suffix when one group has two namings of the configuration or several same-seed repeats), best validation loss, host (superscript: A = the RTX 5090 host, B = the first A6000 host, C = the second A6000 host, ? = host not recorded), and the spread. The 1.5B, 3B and 7B M cells at 3×10^{-4} mix micro-batch decompositions (8×4, 16×2 and 32×1 across their runs); the other cells are decomposition-matched. Seed-to-seed spreads are reported in the text below.

LLM	Width	LR	n	Runs	Spread
1.5B	M	10^{-4}	2	g2 3.221^{B}, g41 3.239^{B}	0.019
1.5B	M	3×10^{-4}	4	g25c/552k 2.713$^{?}$, g39e/anchor 2.712^{B}, g45 2.734^{C}, g45/552k 2.756^{C}	0.044
3B	S	10^{-4}	2	g3 3.233^{B}, g41 3.410^{B}	0.177
3B	S	3×10^{-4}	2	g3v2/552k 2.628$^{?}$, g44 2.798^{C}	0.170
3B	M	10^{-4}	3	g3 3.091^{B}, g41 3.086^{B}, g42 2.985^{B}	0.105
3B	M	3×10^{-4}	5	g25c/552k 2.660$^{?}$, g39b/anchor 2.695^{B}, g3v2/552k 2.710$^{?}$, g44 2.687^{C}, g44/552k 2.652^{C}	0.058
3B	L	10^{-4}	2	g3 2.859^{B}, g41 2.978^{B}	0.119
3B	L	3×10^{-4}	2	g3v2/552k 2.668$^{?}$, g44 2.609^{C}	0.059
3B	XL	10^{-4}	2	g3 2.720^{B}, g41 2.675^{B}	0.045
3B	XL	3×10^{-4}	2	g3v2/552k 2.705$^{?}$, g44 2.803^{C}	0.097
7B	M	10^{-4}	2	g41 3.079^{B}, g42 2.827^{B}	0.252
7B	M	3×10^{-4}	4	g25c/552k 2.492$^{?}$, g39d/anchor 2.476^{B}, g43 2.540^{C}, g43/552k 2.455^{C}	0.085
14B	XS	3×10^{-4}	2	g26 2.511$^{?}$, g39c 2.589^{B}	0.078
14B	S	rule	3	g48/552k 2.479^{C}, g49-rep3/552k 2.488^{C}, g49-rep4/552k 2.507^{C}	0.028
14B	M	10^{-4}	2	g1 3.115^{A}, g2 3.115^{A}	0.000
14B	M	3×10^{-4}	3	g25b/552k 2.522$^{?}$, g26/552k 2.473$^{?}$, g39/anchor 2.515^{B}	0.049
14B	L	10^{-4}	2	g1 3.043^{A}, g7 3.042^{A}	0.001
14B	XL	10^{-4}	2	g1 2.942^{A}, g7 2.940^{A}	0.002

14B L and XL for two seeds (Table 8). The 0.05 threshold recorded before the runs is exceeded at every L and XL cell.

Holding $\eta\lambda$ fixed raises every width whose rule rate is below the anchor's, by $+0.06$ (14B L) to $+0.14$ (14B XL) at all three scales, and lowers the narrow 3B widths (XS -0.036 over two seeds, S -0.061) but not 7B XS ($+0.003$). Relative to the shared 3×10^{-4}, the $\eta\lambda$-fixed rule keeps most of the XS gain at 3B and 7B (2.759 and 2.686 against 2.835 and 2.737) but retains only 0.02–0.05 of XL's 0.13–0.16 gain at 7B and 14B and none of L's at 14B (2.489 and 2.516 against 2.483, i.e. 0.005–0.033 worse). At 3B the $\eta\lambda$-fixed five-width curve is 2.773/2.640/2.695/2.720/2.675 for XS–XL (seed 42), a span of 0.13 with S lowest, against 0.23 under the rule with fixed λ and 0.20 under the shared rate. The rule as evaluated in the main text, with λ fixed, transfers the pair $(\eta, \eta\lambda)$; that is the form whose brackets and held-out predictions we validated, and we recommend it as such.

D TRAINING CURVES

Figure 4 shows the validation-loss trajectories behind the 3B rows of Table 3 under the fixed 10^{-4} and under the width-aware rule. Under the fixed rate the curves separate by width from the first evaluation and the gaps persist to the end of the schedule; every width has flattened over the last 4,000 steps (changes of about 0.003 or less), so the narrow adapters end on a higher plateau rather than still descending. Under the rule the five widths stay within about 0.3 nats of one another throughout training and end within 0.24.

Table 8: Effective-weight-decay control (T=64, D=552K, rule LR): best validation loss with the paper's fixed $\lambda = 0.1$ and with λ rescaled so that $\eta\lambda$ equals the anchor's. The fixed-λ values are the Table 3 runs (seed-123 values from Appendix B); the 3B M anchor is shared by both columns. Shared: the shared-3×10^{-4} cell of Table 3 (for 3B M, the anchor run, 2.695).

LLM	Width	Rule LR	λ_{eff}	Seed	$\lambda = 0.1$	$\eta\lambda$ fixed	Shared
3B	XS	8.26×10^{-4}	0.036	42 / 123	2.786 / 2.803	2.773 / 2.744	2.835
3B	S	4.40×10^{-4}	0.068	42	2.701	2.640	2.713
3B	M	3×10^{-4}	0.100	42	2.695	2.695	2.695
3B	L	2.28×10^{-4}	0.131	42	2.601	2.720	2.639
3B	XL	1.54×10^{-4}	0.195	42 / 123	2.553 / 2.549	2.675 / 2.646	2.754
7B	XS	7.98×10^{-4}	0.038	42	2.683	2.686	2.737
7B	XL	1.55×10^{-4}	0.193	42	2.436	2.551	2.567
14B	L	2.30×10^{-4}	0.130	42 / 123	2.430 / —	2.489 / 2.516	2.483
14B	XL	1.57×10^{-4}	0.191	42 / 123	2.428 / 2.412	2.564 / 2.536	2.588

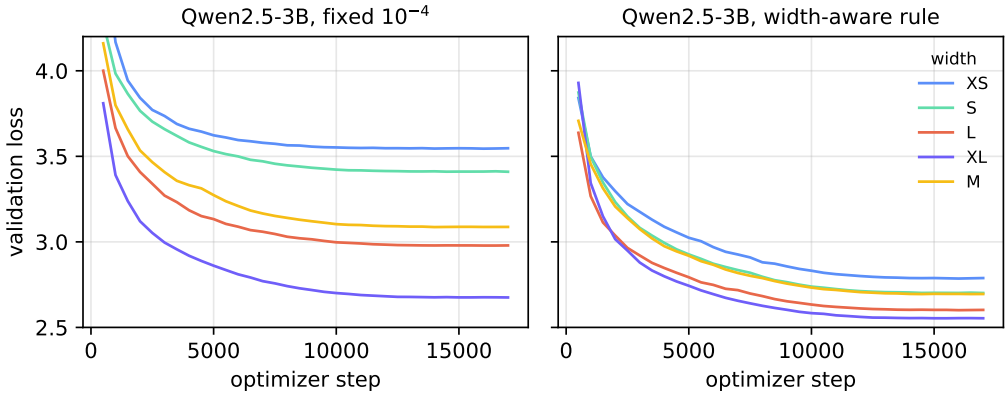


Figure 4: Validation loss every 500 optimizer steps at 3B (T=64, D=552K, seed 42) for the five widths under the fixed 10^{-4} (left) and the width-aware rule (right): one seed-42 run per width (the fixed-LR width-sweep runs and the rule runs, with the M anchor run on the right), so the endpoints are individual runs, not the cell means of Table 3.

Table 9: Best validation loss at T=64, D=552K under two further shared LRs (G37; the two XS cells at 5×10^{-5} are omitted as unverified, marked ^{u} in Table 19). At 2×10^{-4} the widest adapters are best or tied at both scales.

Width	3B, 2×10^{-4}	3B, 5×10^{-5}	14B, 2×10^{-4}	14B, 5×10^{-5}
XS	3.097	—	3.143	—
S	2.953	3.950	3.015	3.555
M	2.707	—	2.855	—
L	2.741	3.355	**2.449**	3.298
XL	**2.642**	**3.048**	2.455	**3.165**

E WIDTH-SPECIFIC LR DIAGNOSTIC AND 14B BRACKETS

Anchor bracket from above. Width M at 4×10^{-4} and 6×10^{-4} reaches 2.804 and 2.994 at 1.5B, 2.697 and 2.854 at 3B, and 2.601 and 2.878 at 7B, against 2.712, 2.695 and 2.476 for the 3×10^{-4} anchor runs (same seed and micro-batch), so the anchor is bracketed from above at every scale, as at 14B (Table 10, where 4×10^{-4} is worse by 0.11). Table 11 lists these and the other control runs of Sections 4.1–5.

Table 10: Per-width LR brackets at 14B (T=64, D=552K). Bold: rule-predicted LR of Eq. 1. For L and XL the sampled minimum is at the rule LR and is bracketed on both sides. S at its rule LR: six runs of the same configuration on two hosts; † one with an unrecovered spike (3.894, first host) and five that trained normally on the second host (2.479, 2.488, 2.507 with seed 42; 2.490 with seed 123; 2.494 with seed 456); XS at 5×10^{-4}: two such runs (2.901, 2.762), both worse than the neighbouring rates.

Width	LRs tested ($\times 10^{-4}$)	Best validation loss
XS	3 / 5 / **7.8**	2.589 / 2.901, 2.762 / 2.563
S	3 / 3.5 / **4.3**	2.697 / 2.559 / 3.894†; 2.479–2.507 (5 runs)
M	**3** / 4	2.515 / 2.622
L	1.6 / **2.3** / 3	2.816 / 2.430 / 2.483
XL	0.8 / **1.6** / 3	2.996 / 2.428 / 2.588

Table 11: Control runs referenced in the main text (T=64, D=552K, best validation loss). Comparator: the Table 3 run(s) the control is read against. The decomposition-matched reruns of the fixed-rate sweeps are in Table 12.

Control	LLM	Width	LR ($\times 10^{-4}$)	Seed	Setting	Loss (comparator)
Anchor above	1.5B	M	4 / 6	42	16×2	2.804 / 2.994 (2.712)
Anchor above	3B	M	4 / 6	42	32×1	2.697 / 2.854 (2.695)
Anchor above	7B	M	4 / 6	42	8×4	2.601 / 2.878 (2.476)
Clipping	14B	XS	7.77	42	clip 10 / off	2.556 / 2.549 (2.563, 2.552)
Heads	14B	XL	1.57	42 / 123	8 heads, dim 192	2.549 / 2.339 (2.428, 2.412)
Heads	14B	XS	7.77	42	2 heads, dim 128 / 8 heads, dim 32	2.539 / 2.549 (2.563, 2.552)
Decay	14B	XS	7.77	42 / 123	$\eta\lambda$ at the anchor's	2.471 / 2.538 (2.563, 2.552)
Decay	14B	S	4.33	42 / 123	$\eta\lambda$ at the anchor's	2.508 / 2.464 (2.492, 2.490)
Clipping	14B	XS	3	456 / 789	clip off	3.068 / 3.068 (3.057, 3.065)
Seeds	14B	XS	3	456 / 789 / 1011	8×4	3.057 / 3.065 / 3.060 (2.495–2.589)

Table 12: Decomposition-matched reruns of the fixed-10^{-4} width sweeps at 7B and 14B (seed 42, 8×4 micro-batch) against the Table 3 runs (7B 16×2; 14B 16×2 on another host). The 7B M cell of Table 3 has two runs.

LLM		XS	S	M	L	XL
7B	rerun 8×4	3.275	3.003	3.087	2.657	2.555
7B	Table 3 runs	3.281	2.995	3.079, 2.827	2.668	2.558
14B	rerun 8×4	3.393	3.230	3.113	3.043	2.941
14B	Table 3 runs	3.119	3.227	3.115	3.043	2.941

Early-training diagnostics at 3B and 14B. To see where the 14B difference sits, we ran 2,000-step diagnostic runs on one host with the full-run scheduler (horizon 17,267 steps, 345 warm-up steps) and recorded the pre-clipping gradient norm of the adapter (clip value 1.0) and the response of the frozen LLM to each logged step, K_t: the mean KL divergence between its next-token distributions before and after the step, evaluated only at caption tokens, on a fixed pool of 128 probe captions split into microbanks of eight that are cycled by step index (the full pool is evaluated at steps 0, 20, 100 and 500). Steps are logged at every optimizer step up to 100, every 10 steps to 500 and every 100 steps thereafter, so the warm-up statistics below cover steps 1–100 at full resolution. Table 13 summarizes the warm-up. At 14B the narrow adapter receives pre-clipping gradients roughly 40–60× larger than the M and XL adapters by median (143–250 against 2–4; maxima above 10^4 against below 90; 9–15 of the first 100 steps exceed 10^3), and the frozen LLM's output moves about 10× more per step for XS (median K_t 0.036–0.039 against 0.003–0.005, and 0.051 for XS at 6×10^{-4}). Clipping at 1.0 is active at every logged warm-up step for every width; after warm-up it remains active for XS (100% of logged steps) but rarely for M (13%) and XL (0%). At 3B the differences are far smaller: gradient medians 22–112 (XS/XL about 5×) and K_t 0.010–0.043 (about 3×). The gradient-norm gap does not reach the parameters: over the same warm-up steps the relative parameter change per step of the 14B XS adapter is *smaller* than that of M and XL (blocks 0.7×10^{-4} against 2.8–3.3 $\times 10^{-4}$; output projection 2.8×10^{-4} against 5.8–7.7 $\times 10^{-4}$; input projection and queries within a factor of two), since Adam and clipping normalize the step; the larger output-distribution change of XS is therefore

not accompanied by larger relative update norms in the measured parameter groups (direction and per-tensor structure of the updates were not compared). After warm-up K_t is 0.002–0.006 for every run. The step-2000 validation losses of these runs are not comparable across runs: two 14B XS runs with identical configuration and seed on the same host reach 3.339 and 2.950, separating during the warm-up from kernel-level numerical differences that training amplifies (Appendix B), so we report only the warm-up statistics, which agree between the two.

Table 13: Warm-up diagnostics of 2,000-step runs at T=64, D=552K, shared LR 3×10^{-4} (seed 42; a second run with seed 123 at 3B and an identical replicate at 14B XS in parentheses). Gradient norm of the adapter before clipping and frozen-LLM response K_t per step, medians over steps 1–100; maxima over the same steps.

LLM	Width	grad norm, median	grad norm, max	steps $> 10^3$	K_t, median
3B	XS	112 (89)	696 (1,744)	0 (1)	0.030 (0.043)
3B	M	34 (28)	408 (199)	0 (0)	0.010 (0.013)
3B	XL	22 (28)	504 (488)	0 (0)	0.031 (0.023)
14B	XS	250 (143)	19,328 (14,080)	15 (9)	0.036 (0.039)
14B	M	4	88	0	0.003
14B	XL	4	38	0	0.005

F Other adapter families

The Q-Former width sweep is reported in Table 4.

MLP projector (fixed capacity, 3B, 10^{-4}). A 2.5M-parameter MLP projector reaches 3.710/3.705/3.703/3.524 at $T = 16/32/64/128$, against 3.099/2.996/2.985/3.119 for the 26M Perceiver (single runs of the fixed-LR token sweep) and 3.063/3.081/3.072/2.956 for a 21M Q-Former at the same LR: the low-capacity projector trails both by 0.4–0.7 nats at every token count. We did not sweep MLP width or evaluate interleaved cross-attention adapters.

G Visual token count

Token sweeps at width M under the selected 3×10^{-4} (best validation loss). Each cell is a single run of the token-sweep suite, so the T=64 cells are individual replicates of the Table 3 cells rather than their means (for example 3B 2.687 here against the five-run mean 2.681; Table 7).

LLM	T=4	8	16	32	64	128	256
1.5B	2.940	2.859	2.946	2.732	2.734	2.850	2.846
3B	2.890	2.746	2.707	2.625	2.687	2.684	2.698
7B	2.668	2.698	2.541	2.546	2.540	2.498	2.563
14B	2.717	2.667	2.431	2.521	2.479	2.461	2.481

Table 14: Token sweeps at width M under the selected 3×10^{-4} (best validation loss; single runs).

At every scale the gain from T=4 to T=16–32 (0.13–0.29 nats) exceeds any change beyond T=32, and the variation over $T \geq 32$ (0.06–0.12) is within the seed and nondeterminism spread of Appendix B, so we read the token axis as saturating by $T \approx 32$ rather than as ordered.

H Held-out validation of width-adjusted loss prediction

Coefficients of the prospective fit. The descriptive fit of Section 4.3, $L = a_s + c\,N_A^{-\gamma}$ with γ free and N_A in millions, fitted to the 19 rule-suite runs available before the held-out runs, has $a_s = 2.6125/2.5373/2.3824/2.3660$ for 1.5B/3B/7B/14B, $c = 0.4525$ and $\gamma = 0.4495$ (the 19 runs are the seed-42 rule and anchor runs of the four scales excluding the failed 14B S run); with the rule LR of Eq. 1 it reproduces the recorded d=896 predictions 2.454 (14B) and 2.629 (3B) to three

decimals (the applied rates were 2.605 and 2.594×10^{-4}; Eq. 1 with the exact parameter counts gives 2.604 and 2.589×10^{-4}). The d=1280 predictions used the same form refitted to those 19 runs plus the two d=896 runs ($a_s = 2.6130/2.5466/2.3829/2.3579$, $c = 0.4491$, $\gamma = 0.4483$), which reproduces the recorded 2.423 (14B) and 2.449 (7B) to three decimals.

Table 15: Held-out prediction RMSE with the rule's exact coordinate $x = \eta\sqrt{N_A}^{\,\alpha}$ (same cells, splits and fitting procedure as Table 5).

		leave-one-width-out			leave-one-LR-out		
LLM	points	α=0	α=1	α_{fit}	α=0	α=1	α_{fit}
1.5B	16	0.289	**0.129**	0.206	0.185	**0.084**	0.123
3B	24	0.227	0.093	**0.088**	0.167	0.099	**0.070**
7B	16	0.180	**0.085**	0.095	0.214	**0.082**	0.105
14B	30	0.219	0.167	**0.158**	0.181	**0.153**	0.162

Held-out prediction. The in-sample residuals of Table 16 share runs with the fit. Here the same primary analysis set (T=64, D=552K) is first reduced to one point per (width, LR) by averaging the runs of each cell, so that no replicate sits on both sides of a split; then one width (or one learning rate) is held out, a quadratic in $\log(\eta\, d^{\alpha})$ is fitted on the rest, and the held-out points are predicted. α_{fit} is the exponent with the lowest in-sample residual on the fitting part of each split. Table 5 gives the prediction RMSE averaged over splits.

Table 16: Residual RMSE (nats) of a quadratic-in-$\log x$ fit of validation loss against $x = \eta\, d^{\alpha}$, per LLM, over the runs of the primary analysis set of Section 3.3 (T=64, D=552K; all widths, all learning rates). Single-host seed spread and within-cell replicate spread are reported in Appendix B (replicates reach 0.10–0.25, Table 7). Bold: best α. At its best exponent the 14B residual is 1.5–2× the 1.5B–7B level.

LLM	n	α=0	0.25	0.5	0.75	1.0	1.25	1.5
1.5B	23	0.204	0.154	0.110	**0.091**	0.095	0.113	0.136
3B	52	0.194	0.144	0.093	**0.071**	0.085	0.110	0.134
7B	26	0.149	0.105	**0.087**	0.089	0.091	0.098	0.107
14B	52	0.171	0.149	**0.138**	0.146	0.171	0.201	0.227

Table 17: Held-out prediction RMSE of Table 5 with the 14B S run that showed an unrecovered spike included at its recorded loss (3.894); the S cell mean becomes 2.726 over six runs and no other loss cell changes. The 32B failure (5.65 at 3×10^{-4}) lies outside the 1.5B–14B range of this table and is reported in Appendix Q.

		leave-one-width-out			leave-one-LR-out		
LLM	points	α=0	α=1	α_{fit}	α=0	α=1	α_{fit}
1.5B	16	0.289	**0.145**	0.184	0.185	**0.096**	0.121
3B	24	0.227	0.110	**0.097**	0.167	0.110	**0.076**
7B	16	0.180	**0.093**	0.096	0.214	**0.086**	0.108
14B	30	0.222	0.184	**0.174**	0.193	**0.165**	0.185

Fixing $\alpha = 1$ lowers held-out error relative to $\alpha = 0$ in every model and split type: by 34–60% at 1.5B–7B (1.5B 50/48%, 3B 51/34%, 7B 48/60% for width-out/LR-out) and by 17/10% at 14B. Using the rule's exact coordinate $\eta\sqrt{N_A}$ instead of ηd lowers the in-sample residual slightly at every scale (1.5B 0.100 against 0.105, 3B 0.086 against 0.099, 7B 0.067 against 0.068, 14B 0.153 against 0.165), so the approximation does not drive the 14B result; these in-sample values are computed on configuration means, whereas Table 16 and Figure 2 use individual runs, which is why the ηd residual reads 0.165 here and 0.171 there. Table 15 repeats the held-out test in the exact coordinate: $\alpha = 1$ reduces error relative to $\alpha = 0$ by 41–61% at 1.5B–7B and by 15–24% at 14B, slightly more than with ηd at every scale, so the approximation is not what separates 14B. Table 17 repeats the held-out

test with the 14B S run that did not train included at its recorded loss: only the 14B S cell changes (its mean becomes 2.726 over six runs); for the prespecified $\alpha = 1$ predictor the 14B held-out RMSE changes by 0.001 (0.183 to 0.184, width-out) and 0.002 (0.163 to 0.165, LR-out), the largest change across all predictors is 0.023 (α_{fit}, LR-out, 0.162 to 0.185), and the run-level in-sample residual of Table 16 rises from 0.138 to 0.232 at $\alpha = 0.5$ because a single run at 3.894 enters the fit. The improvement of the prespecified width-adjusted predictor over η alone therefore persists when the failed run is included, whereas the run-level in-sample residual is markedly more sensitive to the inclusion; we report both. Selecting the exponent on the fitting data (α_{fit}, chosen by in-sample residual on the fitting part of each split, not on an independent development set) improves 3B (0.110 to 0.097 and 0.076) and 14B width-out, and worsens 1.5B and 7B LR-out, with no consistent gain. At 14B the error remains 0.16–0.18 for $\alpha = 1$ and α_{fit}, larger than at 1.5B–7B, and adding width to the coordinate reduces error by only 10–17% against 34–60%. The in-sample residuals (Table 16) share runs with the fit. RMSE here is the arithmetic mean over splits of the per-split RMSE; leave-one-LR-out removes one exact learning-rate value at a time, and since the other cells still carry both η and d, this tests prediction at a held-out rate (interpolation for interior rates, extrapolation for the extreme ones), not a new protocol.

I Training-log spike scan

For the 132 runs with training logs (loss every 50 optimizer steps) we flag a spike when loss exceeds the rolling median of the previous 20 logged points (at least 6, i.e. 300 steps of history; earlier points are not assessed) by more than $3 \times 1.4826\,\mathrm{MAD}$ and by more than 1 nat, and record whether loss returns to within 0.2 of the median within 200 steps. Because the detector needs 300 steps of history, it cannot flag spikes inside the 345-step warm-up; as a coarser check, the warm-up maximum exceeds the first logged loss (step 50) by more than 0.3 nats in 5 of the 159 runs whose logs cover the warm-up (132 of them have complete logs and enter the spike scan above), all of them 3B width S (by 0.30–0.73), and by less in every other run. Post-warm-up spikes occur in four 14B runs: S at 4.3×10^{-4} (steps 1150–1700, peak 6.53, not recovered, final 3.89; five further runs of the same configuration on another host showed no spike and reached 2.479–2.507), XL at 1.6×10^{-4} seed 123 (3150–3300, peak 5.08, recovered, best 2.412), XL at 3×10^{-4} (3750, peak 4.04, recovered, best 2.588) and ML at 2.6×10^{-4} (6400, peak 4.56, recovered, best 2.412). No 1.5B, 3B or 7B run spikes. Spikes shorter than 50 steps are invisible at this logging resolution.

J Compute

Runs use one NVIDIA A6000 (48 GB) or, for a subset of the earlier groups, one RTX 5090 (32 GB), except the 32B runs, which use two A6000s; a seed-matched host comparison at 14B differs by 0.035, within seed noise, whereas some 3B cells differ by up to 0.17 between hosts (Table 7). Wall time at T=64, D=552K on an A6000 (median of logged runs): 1.5B 3.8 h and 3B 5.1 h at 32×1, 7B 7.7 h at 16×2 and 14B 13.9 h at 8×4 (4–16 h across token counts, micro-batch settings and hosts), scaling roughly with $T + 56$. The last partial training batch is dropped (552,544 pairs consumed). Validation uses shuffled batches and drops the final incomplete batch, retaining 5,568 pairs at batch sizes 16 and 32 or 5,576 pairs at batch size 8. Effective batch is 32 throughout; micro-batches of 16×2 (7B), 8×4 (14B, and 3B/1.5B at T=256) and 4×8 (7B at T=256) preserve the effective batch size and the number of optimizer updates while fitting memory (the memory bottleneck at long prefixes is the batch × sequence × $|V|$ logits, $|V| \approx 152$K, rather than activations). Each micro-batch contributes its token-mean loss divided by the number of micro-batches, so when micro-batches differ in token count the objective can weight tokens differently from a full-batch token mean, and the numerical trajectory is not identical across decompositions. The decomposition is uniform within most comparisons but not all: at 1.5B and 3B the width tables use 32×1 throughout except the M-width LR-sweep runs (8×4) and the 1.5B rule runs (16×2); at 7B the fixed-rate runs use 16×2 and the selected-rate and rule runs 8×4 or 16×2; at 14B the fixed-rate runs (RTX 5090 host) use 16×2 and every other T=64 Perceiver width and bracket run 8×4 (seed replicates at M use 16×2 and the T=256 run 4×8). The fixed-versus-selected comparisons at 7B and 14B therefore also differ in micro-batch decomposition. A decomposition-matched rerun of the fixed-rate 7B and 14B width sweeps at 8×4 (seed 42) reproduces eight of the ten cells within 0.012 (14B S–XL within 0.003 despite a different host); the exceptions are 14B XS (3.393 against 3.119, a single earlier run) and 7B M (3.087 against

3.079 and 2.827, a cell whose two same-configuration runs already differ by 0.25); the two exceptions do not isolate decomposition from run-to-run and host variation (Table 12). The 14B bracket and rule-versus-shared comparisons and the same-seed repeats of Table 7 are decomposition-matched, except the three M cells noted there. The runs reported here total roughly 4,000 A6000 GPU-hours (timed from logged step rates where a log exists, otherwise estimated from runs of the same LLM and token count; 32B runs counted on two GPUs); the 44 two-thousand-step diagnostic and sensitivity runs (Table 13, Appendix B) account for about 60 of the total.

K Inference cost of visual tokens

Prefill latency, decode throughput and peak memory on one A6000 (batch 1, greedy decoding of 64 tokens, width M). $T{=}64$ costs almost nothing over $T{=}32$; at 14B the prefill penalty grows to $+25\%$ at $T{=}128$ and $+89\%$ at $T{=}256$, while decode throughput stays within 2% and peak memory within 0.1 GB.

Table 18: Inference cost of visual tokens at width M on one A6000 (batch 1).

LLM	T	Prefill (ms)	Generate 64 tok (ms)	Decode (tok/s)	Peak mem (GB)
3B	32	37.6	1621	40.4	6.74
3B	64	37.9	1632	40.1	6.74
3B	128	38.8	1639	40.0	6.75
3B	256	45.6	1663	39.6	6.76
14B	32	71.8	3128	20.9	28.56
14B	64	70.4	3148	20.8	28.56
14B	128	89.6	3175	20.7	28.58
14B	256	136.0	3255	20.5	28.62

L Licenses of existing assets

Qwen2.5-1.5B/7B/14B: Apache-2.0; Qwen2.5-3B: Qwen Research License (research use); SigLIP-SO400M/14-384: Apache-2.0; LLaVA-Pretrain-558K: per-image terms inherited from CC3M, LAION and SBU, used as released by the LLaVA authors without redistributing images; VQAv2 and TextVQA annotations: CC-BY-4.0; COCO captions: CC-BY-4.0 with Flickr terms for images.

M Experiment inventory

LoRA (G6) fine-tunes the LLM with rank 64 alongside the adapter at 10^{-4}; at 3B (four unverified runs, ^{u} in Table 19) it reaches 2.694–2.712 across $T \in \{16, 32, 64, 128\}$, against 3.087 for the frozen LLM at the same rate. We do not analyse it further because the frozen-LLM premise of this paper does not hold there.

Table 19: Run groups and how this paper uses them.

Group	Purpose	Runs	Used in
G1, G41	fixed-LR width sweep, 4 scales	20	Table 3, Fig. 1, fit of Section 4.2, Table 22
G2, G42	fixed-LR token sweep	35	the T=64 runs enter Table 3 and the fit of Section 4.2; two checkpoints (T=8, 128) enter Table 22
G3, G3v2	tokens × width at 3B, both LRs	32	App. N; the T=64 runs also enter Table 3 and the fit
G4, G4v2	input resolution, both LRs	15 (+5[u])	App. O; the two 384-px T=64 runs enter the fit and the 224-px T=32 run Table 22; the selected-rate 224-px runs are [u]
G6	LoRA on the LLM	4 (+8[u])	App. M note (the 3B values there are [u] runs)
G8	adapter depth	10	App. P
G9, G9v2	seed repeats, both LRs	30	App. B; the T=64 runs enter the fit
G11	MLP / Q-Former token sweep	8	App. F
G16	0.5B LR sensitivity	24[u]	App. Q
G25b/c, G35	LR sweep at M, 4 scales	11 (+1[u])	Table 2; the 3×10^{-4} runs enter the M cells of Table 3 and the fit
G26, G40, G39c, G43–G47	selected-LR width and token sweeps, 4 scales	73	Tables 3, 7, App. G, fit of Section 4.2
G34	Q-Former width sweep, both LRs	20	Table 4
G37	width-specific LR diagnostic	16 (+2[u])	App. E, fit of Section 4.2
G39 series	width-aware rule, brackets, held-out, seeds	32 (+5[u])	Table 3, Sec. 4.3, fit; the S run with the unrecovered spike is kept as a record and excluded from cell means
G32/G33	32B LR sweep at 2×10^{-4}, 5×10^{-5}	10	App. Q
G50	32B at 3×10^{-4}	1	App. Q (failed to learn)
A1, A1c, A1d, A2s, G48, G49	diagnostic runs, sensitivity probes and 14B replicates	50	App. E, Table 13; G48/G49 enter the 14B S cell of Table 3 and the fit
G0v2, G7	data-scaling and long-token groups; their T=64, D=552K runs enter Table 3 (14B L/XL, with G1) and the fit, two serve as downstream checkpoints	6 (+11[u])	Table 3, fit, Table 22
G5	data-scaling runs at 3B; five T=64 runs enter the fit	5 (+9[u])	fit of Section 4.2
W1–W3	weight-decay control, rule-cell seed replicates, deterministic pair	14	App. C, B
X1–X3	effective-decay controls at 7B, 14B and 3B S/L (X4: downstream evaluation of Table 24)	8	App. C, S
Y1–Y5	anchor bracket from above, decomposition-matched fixed sweeps, clipping and head-count controls, 14B XS seeds	22	Sec. 4.1, 5; App. B, E, J
Z1–Z4	effective-decay and head-configuration controls at 14B XS/S, clipping off at the shared rate, head-count replicate at XL	9	Sec. 5; App. E, B
SFT	instruction tuning and generation evaluation of the fifteen 3B width-sweep checkpoints	15	App. T

[u] exploratory or unverified runs (the latter have ledger `verified = False`), shown only in the listed appendix or omitted, and used in no primary result or claim.

N TOKEN COUNT AND WIDTH INTERACTION

At 3B, we evaluate a 4×4 grid over $T \in \{8, 32, 64, 128\}$ and widths S/M/L/XL at both shared learning rates (32 runs; each cell is a single run, so the 3B M T=64 cell at 3×10^{-4}, 2.710, is one of the five replicates behind the Table 3 mean 2.681). Under 10^{-4}, loss decreases monotonically with width at each token count (XL is best, with a spread of 0.34–0.64). Under 3×10^{-4}, the spread across widths at each token count shrinks to 0.08–0.20 and the ordering is not monotone, reproducing the protocol dependence of Section 4.1. Within a width the token effect is small at both rates (at most 0.3 nats from T=8 to 128, and at most 0.2 outside the S column).

	fixed 10^{-4}				selected 3×10^{-4}			
T	S	M	L	XL	S	M	L	XL
8	3.352	3.045	2.920	**2.830**	2.924	2.810	**2.726**	2.756
32	3.394	3.087	2.917	**2.752**	2.782	**2.682**	2.700	2.734
64	3.233	3.091	2.859	**2.720**	**2.628**	2.710	2.668	2.705
128	3.235	2.991	2.942	**2.894**	2.750	2.710	2.655	**2.651**

Table 20: Token count × width at 3B under both shared learning rates (best validation loss; single runs).

O INPUT RESOLUTION

Changing the encoder input from 384 to 224 pixels changes the number of encoder patch tokens from 729 to 256 while T is held fixed (3B, width M); the two settings use the 224- and 384-pixel SigLIP checkpoints, so the encoder weights change with the resolution. Under 10^{-4} the two resolutions differ by at most 0.10 nats (224 px slightly better at $T \geq 32$); under 3×10^{-4} the differences are 0.01–0.12 with inconsistent sign (224 px better at T=16 and 128, 384 px better at T=32). We therefore do not treat the compression ratio T/T_0 as a scaling axis.

	fixed 10^{-4}		selected 3×10^{-4}	
T	224 px	384 px	224 px	384 px
8	3.140	3.114	2.751	2.781
16	3.103	3.109	2.670	2.717
32	3.069	3.088	2.727	2.608
64	3.081	3.113	2.690	2.676
128	3.020	3.118	2.659	2.756

Table 21: Input resolution (384 vs. 224 px) at 3B, width M, under both shared learning rates. The five selected-rate 224-px runs are unverified (^{u} in Table 19) and enter no other result.

P ADAPTER DEPTH

With width M and T=64 at 10^{-4}, one block reaches 3.038 (3B) and 3.097 (7B); four blocks 2.970 and 2.790; six blocks 2.965 and 2.790. The corresponding parameter counts at 3B are 14M, 50M and 73M for depths 1, 4 and 6. Increasing depth from four to six changes loss by at most 0.005. The main study fixes depth at two and varies width rather than selecting an optimal depth.

Q BOUNDARY CASES: 0.5B AND 32B

Qwen2.5-0.5B is excluded from the fitted range. Over 24 runs at $T \in \{32, 64, 128\}$, learning rates 10^{-5} to 10^{-4} and two seeds, 10^{-4} is the best tested rate for every token count and seed (best validation loss 5.6–6.0 at 10^{-5}, 4.1–4.8 at 3×10^{-5}, 3.9–4.6 at 5×10^{-5}, 3.37–3.84 at 10^{-4}), the two seeds differ by up to 0.72 at the same setting (0.45 at 10^{-4}), and three runs end 0.4–1.7 above their best. Its runs appear only in this appendix and in Appendix S, and enter no other result.

At 32B, the M run at 2×10^{-4} reached a best validation loss of 3.328 but ended at 4.983, and the XS run at that rate 3.326 and 6.021; the single run at 3×10^{-4} (width M) never fell below 5.650. These

observations do not establish a stability boundary or a failure frequency; the 32B runs enter no other result.

R BROADER IMPACTS

This paper studies how to compare and train adapters between public frozen models; it introduces no new generative, surveillance or data-collection capability beyond the public backbones. Its practical effect is to reduce wasted compute: width comparisons under a shared learning rate put the widths at different stages of optimization, and the one-anchor rule removes the per-width search. The reported experiments consumed several thousand A6000-hours; we disclose the run inventory so that others need not repeat it.

S DOWNSTREAM NLL

Table 22: Pretraining validation loss against ground-truth-answer NLL (VQAv2, TextVQA) and caption NLL (COCO) for 11 checkpoints. Bottom rows: Pearson and Spearman correlation of each column with validation loss.

Group	LLM	Width	T / D	LR	Val loss	VQAv2	TextVQA	COCO
G7	7B	XL	128 / 552K	10^{-4}	2.547	7.930	6.348	3.403
G41	3B	XL	64 / 552K	10^{-4}	2.675	7.661	5.812	3.521
G0v2	7B	M	64 / 2.2M	10^{-4}	2.703	8.550	7.075	3.539
G3	3B	XL	64 / 552K	10^{-4}	2.720	7.734	6.060	3.620
G42	7B	M	128 / 552K	10^{-4}	3.053	7.333	6.898	3.715
G4	3B	M	32 / 552K (224px)	10^{-4}	3.069	9.103	7.562	3.639
G41	3B	M	64 / 552K	10^{-4}	3.086	9.004	7.075	3.662
G1	14B	M	64 / 552K	10^{-4}	3.115	9.339	7.958	4.012
G2	0.5B	M	8 / 552K	10^{-4}	3.383	10.529	8.155	3.679
G3	3B	S	32 / 552K	10^{-4}	3.394	9.275	8.374	3.720
G41	3B	S	64 / 552K	10^{-4}	3.410	9.160	7.904	3.965
Pearson r vs. val loss						0.73	0.88	0.72
Spearman ρ vs. val loss						0.77	0.87	0.88

On 11 checkpoints spanning 0.5B–14B, widths S/M/XL and $T \in \{8, 32, 64, 128\}$, validation loss correlates with ground-truth-answer NLL at Pearson 0.73 (VQAv2 (Goyal et al., 2017), 214,354 questions) and 0.88 (TextVQA (Singh et al., 2019), 5,000), and with caption NLL at 0.72 (COCO (Chen et al., 2015), 5,000 validation images); Spearman 0.77/0.87/0.88. These correlations hold across the mixed pool of scales; within the three 7B checkpoints only the COCO ordering follows validation loss. The evaluation scores the ground-truth answer or caption under the pretraining-only adapter (for VQAv2 and TextVQA the most frequent of the human answers to each question; for COCO the first reference caption of each image); free-form generation is uninformative for adapters that were never instruction-tuned, and task accuracy after instruction tuning is evaluated at 3B in Appendix T (Section 5).

Protocol comparison at 3B. To test whether the protocol effects of Table 3 carry beyond validation loss, we scored the fifteen 3B width-sweep checkpoints of the three protocols on the same three tasks (Table 23; VQAv2 on its first 20,000 questions, so its absolute values are not comparable with Table 22). The large effects transfer. Moving XS, S, M or L from the fixed 10^{-4} to either higher-LR protocol lowers VQAv2 NLL by 0.4–3.0 and TextVQA NLL by 0.7–2.3, in the same direction as validation loss at every one of these widths, and the XS–XL downstream gap collapses with the validation gap (VQAv2 3.05 under the fixed rate against 0.26 and 0.54 under the shared rate and the rule; TextVQA 2.86 against 0.21 and 0.36). The small differences do not: among the ten checkpoints trained at 3×10^{-4} or the rule LR, which lie within 0.28 nats of validation loss, the rank correlation between validation loss and downstream NLL is weak (Spearman -0.24/0.19/0.15 for VQAv2/TextVQA/COCO, against 0.54/0.70/0.61 over all fifteen), and the best width by VQAv2 or TextVQA is not the best by validation loss under either protocol. The M checkpoints of the shared-rate

and rule rows are same-configuration replicates (validation loss 2.687 and 2.695) yet differ by 0.67 in VQAv2 NLL and 0.30 in TextVQA NLL, as large as typical width-to-width differences within a row. The widest adapter under the fixed 10^{-4} has the lowest TextVQA NLL and the second-lowest VQAv2 NLL of the fifteen despite a validation loss 0.12 nats above its rule-LR counterpart. Downstream NLL therefore supports the protocol-level conclusions of Section 4.1 but not fine width rankings, consistent with the replicate spread of Appendix B.

Table 23: Downstream NLL of the fifteen 3B width-sweep checkpoints (T=64, D=552K, seed 42) under the three protocols: validation loss, ground-truth-answer NLL on the first 20,000 VQAv2 questions and on TextVQA (5,000), and caption NLL on COCO (5,000). Same evaluation as Table 22 except for the VQAv2 sample.

Protocol	Width	Val loss	VQAv2	TextVQA	COCO
fixed 10^{-4}	XS	3.545	10.731	8.668	3.880
	S	3.410	9.165	7.904	3.965
	M	3.086	9.019	7.075	3.662
	L	2.978	10.199	7.541	3.701
	XL	2.675	7.676	5.812	3.521
shared 3×10^{-4}	XS	2.835	7.700	6.416	3.550
	S	2.798	8.189	6.601	3.548
	M	2.687	8.345	6.349	3.717
	L	2.609	8.168	6.258	3.570
	XL	2.803	7.441	6.202	3.685
width-aware rule	XS	2.786	8.659	6.620	3.699
	S	2.701	8.788	6.203	3.606
	M	2.695	7.680	6.049	3.673
	L	2.601	8.523	6.359	3.580
	XL	2.553	8.121	6.264	3.505

Protocol comparison at 7B and 14B. Table 24 repeats the check for the 7B and 14B checkpoints of Table 3 (same evaluation; the 14B fixed-rate row comes from another host and micro-batch decomposition, Appendix J). At 14B the protocol effect transfers at every width: moving off the fixed 10^{-4} lowers VQAv2 NLL by 0.6–3.1 and TextVQA NLL by 0.8–2.7, and over the fifteen checkpoints validation loss correlates with answer NLL (Spearman 0.84 for VQAv2, 0.88 for TextVQA) but not with caption NLL (0.22). At 7B it does not: XS has the lowest or tied-lowest VQAv2 NLL under every protocol (S is lowest for TextVQA under the rule), and over the fifteen checkpoints VQAv2 NLL anti-correlates with validation loss (-0.69), TextVQA is uncorrelated (-0.09), and only caption NLL follows validation loss (0.65); the fixed-rate deficit shows in TextVQA at XS, S and XL (-0.5 to -1.2) but not at M or L. In most shared-rate and rule rows the best width by answer NLL differs from the best by validation loss (exceptions: 14B shared rate, M for both; 14B rule, XL for validation and TextVQA), and the XS–XL downstream gaps do not shrink with the validation gap. Ground-truth-answer NLL of a pretraining-only adapter is therefore a coarse check: it registers the half-nat deficit of the fixed rate at 3B and 14B, not width differences, and at 7B its response depends on the task.

T TASK ACCURACY AFTER INSTRUCTION TUNING

To test whether the protocol effects of Table 3 survive instruction tuning, we fine-tuned the fifteen 3B width-sweep checkpoints of the three protocols with one fixed recipe and scored the resulting models by generation. The recipe: 20,000 LLaVA-Instruct conversations (Liu et al., 2023) and 20,000 VQAv2-train short-answer pairs (Goyal et al., 2017), one pass in 2,500 steps of 16 pairs (micro-batches of 4 with 4 accumulation steps), loss on assistant tokens only; the adapter, initialised from the pretraining checkpoint, at learning rate 10^{-4}, and LoRA (rank 16, α=32) on the query, key, value and output projections of the otherwise frozen LLM, also at 10^{-4}; cosine decay with 50 warm-up steps, weight decay 0.1, seed 42. The recipe was fixed on the selected-rate M checkpoint, chosen among three candidates by VQAv2 accuracy, before the other fourteen runs. Evaluation: greedy decoding with the prompt "Question: {question}\nShort answer:" and at most 16 new tokens

Table 24: Downstream NLL of the 7B and 14B width-sweep checkpoints of Table 3 under the three protocols (validation loss; VQAv2 answer NLL on the first 20,000 questions; TextVQA answer NLL, 5,000; COCO caption NLL, 5,000). The 7B and 14B shared-rate M entries are the g43 and g26 runs; the rule M entries are the anchor runs.

		7B				14B			
Protocol	Width	Val	VQAv2	TextVQA	COCO	Val	VQAv2	TextVQA	COCO
fixed 10^{-4}	XS	3.281	6.89	6.72	3.62	3.119	9.75	8.15	3.91
	S	2.995	7.17	6.76	3.64	3.227	9.49	7.84	4.17
	M	3.079	7.64	7.16	3.72	3.115	9.34	7.96	4.01
	L	2.668	8.27	6.84	3.47	3.043	9.31	7.85	3.89
	XL	2.558	9.06	7.13	3.31	2.942	8.96	7.93	3.87
shared 3×10^{-4}	XS	2.737	7.47	5.99	3.40	2.589	9.16	7.35	3.76
	S	2.620	7.67	6.17	3.31	2.697	7.54	6.40	3.81
	M	2.455	9.34	7.02	3.36	2.473	6.63	5.55	3.97
	L	2.551	8.76	6.74	3.41	2.483	7.62	6.12	3.89
	XL	2.567	8.10	6.63	3.37	2.588	8.22	6.63	3.84
width-aware rule	XS	2.683	6.87	5.66	3.37	2.563	6.62	5.89	3.87
	S	2.504	6.87	5.56	3.32	2.479	6.72	5.37	3.83
	M	2.476	9.68	7.50	3.38	2.515	6.95	5.64	3.92
	L	2.442	9.01	6.80	3.40	2.430	7.35	6.19	3.81
	XL	2.436	8.76	6.61	3.25	2.428	6.86	5.24	3.91

on the first 1,000 questions of VQAv2 val and of TextVQA val, scored with the VQA accuracy metric (a prediction receives $\min(1, m/3)$, where m is the number of the ten reference answers it matches after lower-casing, punctuation and article removal, and whitespace normalization). The selected-rate and rule rows share the M anchor, and a second run of that configuration (the anchor check of Section 4.3) provides the replicate pair shown in the M column.

Table 25: VQA accuracy (%) after instruction tuning of the fifteen 3B width-sweep checkpoints of Table 3 with one recipe (1,000 questions per task; the M column of the lower two rows shows the two same-configuration anchor runs, the selected-rate run / the anchor-check run of Section 4.3).

Protocol	Task	XS	S	M	L	XL	XS−XL
fixed 10^{-4}	VQAv2	54.4	58.1	61.4	62.1	65.4	−11.0
	TextVQA	9.9	11.4	17.6	19.4	22.9	−13.0
selected 3×10^{-4}	VQAv2	65.2	65.6	66.4 / 66.7	69.0	65.4	−0.2
	TextVQA	21.7	26.8	30.7 / 29.3	32.9	21.1	+0.6
width-aware	VQAv2	66.8	69.4	(anchor)	67.7	68.8	−2.0
	TextVQA	28.9	31.2	(anchor)	33.2	32.8	−3.9

The fixed-rate deficit survives instruction tuning. Under the fixed 10^{-4}, XS–L trail the selected rate by about 5–11 VQAv2 and 12–15 TextVQA points and the rule by about 5–12 and 12–20, and the fixed-rate XS–XL gap of 11.0 (VQAv2) and 13.0 (TextVQA) points closes to -0.2 and $+0.6$ under the selected rate. XL at the selected rate is the one poor cell of the two higher-rate rows (65.4 / 21.1, at the level of the fixed-rate XL), matching its over-stepping in pretraining loss (Section 4.3); the rule's XL recovers to 68.8 / 32.8, and the rule's XS and S are 1.6 / 7.2 and 3.8 / 4.5 points above the selected rate's. Coarse orderings transfer: over the fifteen checkpoints, each checkpoint's own pretraining validation loss ranks accuracy at Spearman -0.83 (VQAv2) and -0.92 (TextVQA), and at -0.71 and -0.92 over the ten higher-rate checkpoints. Small within-row differences remain uncertain, given the replicate differences of 0.2 and 1.4 points, as for answer NLL above. The fine-tuning validation NLL on a held-out mix of 500 pairs spans only 1.278–1.314 over the fifteen models, while generation accuracy still separates the protocols. Scope: one scale, one recipe and seed, 1,000 questions per task; the fifteen fine-tuning runs and their evaluations add about 20 A6000 GPU-hours.