Grafts Add Up: Scoring Every Layer Swap in One Forward–Backward Pass

Anonymous authors
Paper under double-blind review

ABSTRACT

Layer swapping combines two models fine-tuned from the same base by replacing some layers of one model with layers of the other. Choosing which layers to swap usually requires building and testing many merged models. We study whether the effect of each donor layer can instead be scored on the host model's own trajectory. The scores add across swapped layers; the error is second order when both experts stay near their shared base. SUTURE uses these scores to rank layer windows from one forward and backward pass per probe, then uses a fixed number of built models to calibrate a language constraint and check its choice. Controlled stacks support the approximation for small expert changes and show where its ranking weakens. On public multilingual models, selected windows improve a teacher-forced answer measure over the host but do not surpass published windows. Their effect on the language of free generation remains unmeasured. The method offers a way to narrow layer-swap search while keeping the final choice subject to direct validation.

1 INTRODUCTION

Layer swapping combines experts trained from a shared base by replacing a band of one model's layers with layers from another. It can transfer mathematical reasoning to a language without task data (Bandarkar et al., 2025), and has also been studied for long reasoning and multilingual safety (Lasbordes et al., 2026; Shin & Hwang, 2026). The choice of layers matters: a swap may import a skill while changing the host's output language.

Existing selection methods use a sweep or search over built candidates (Lasbordes et al., 2026; Akiba et al., 2025), train a predictor on merges (Akizuki et al., 2025), or rank layers using parameter statistics (Bandarkar et al., 2025; Shin & Hwang, 2026). The last approach avoids candidate builds but does not use a task or language readout. We ask whether a single pass through the host can provide readout-specific scores for every possible window before any merged model is built.

Our starting point is to evaluate each donor layer on the states produced by the host. This gives one output difference per layer, independent of the other swaps. Multiplying that difference by the host readout gradient gives a score for the layer. For smooth layers and small grafts, the scores add across a set of swaps with a bounded remainder (Section 3). Thus any window can be scored by adding its layer scores. The first-order attribution itself is related to weight patching (Sun et al., 2026); here we use the additive scores to choose among windows under a language constraint.

SUTURE scores task and language readouts, calibrates the language constraint on built windows, and checks the selected window directly (Section 4). We test the approximation and selection procedure in controlled stacks where every window can be measured, then evaluate public multilingual specialists with a narrower teacher-forced measure (Sections 5 and 6). The readout does not establish the language of free generation.

Contributions. We give a first-order decomposition for arbitrary graft sets, a window-selection procedure with a fixed build budget, and controlled tests that show both useful ranking and its limits. The public-model study checks how much of this account transfers to multilingual specialists while reporting where the current evidence ends.

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

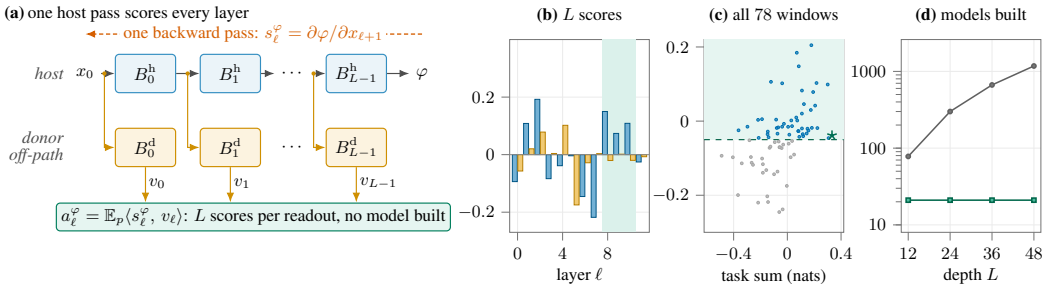

Figure 1: Scoring and selecting a layer window. (a) Donor layers are evaluated at cached host states; one backward pass gives the gradients used in the scores. (b) Task (blue) and language (orange) scores in nats in the seeded stack; the selected window is shaded. (c) Task score (horizontal) against language score (vertical), with feasible choices in blue, infeasible choices in grey, and the selection marked by a star. (d) Merged-model builds for a sweep (grey) and the full SUTURE procedure (green).

2 CHOOSING A BAND IS A CONSTRAINED PROGRAM

Choosing a layer window requires balancing the capability imported from the donor against the host behavior we want to keep. Let a decoder apply L residual units to a state $x_\ell \in \mathbb{R}^{T\times d}$,

$$x_{\ell+1} \;=\; x_\ell + B_\ell(x_\ell;\theta_\ell), \qquad \ell = 0,\dots,L-1, \tag{1}$$

with embedding and unembedding fixed. A host expert θ^{h} and a donor expert θ^{d} are fine-tuned independently from a shared base θ^{b}. As in Bandarkar et al. (2025), the host is the expert whose surface behavior we keep and the donor is the expert whose internal capability we import; in the cross-lingual case the host speaks the target language and the donor reasons.

For $S \subseteq \{0,\dots,L-1\}$, the *graft* θ^S takes donor parameters at the layers in S and host parameters elsewhere. We call any such S a *graft set*, a contiguous one a *window*, and the window a practitioner finally swaps a *band*; layer swapping searches over windows. The theory holds for any partition into sequential units, but units are whole layers here.

Two scalar readouts are computed from the final state at the last prompt position. The *task* readout φ_{task} measures the imported capability, the *language* readout φ_{lang} how strongly the host keeps its output language. Both are differentiable next-token log-probability functionals that need no human annotation (Section 4). Each has its own probe set and may depend on the probe through its answer tokens; we write φ for φ_p. With $\Delta\varphi(S) = \mathbb{E}_p[\varphi(x_L^S(p)) - \varphi(x_L^{\mathrm{h}}(p))]$, where the expectation runs over the readout's probes, the selection problem is

$$\max_{S\in\mathcal{S}} \; \Delta\varphi_{\mathsf{task}}(S) \qquad \text{subject to} \qquad \Delta\varphi_{\mathsf{lang}}(S) \;\geq\; -\tau, \tag{2}$$

with $\mathcal{S}$ the windows or the full power set. A sweep solves (2) by building a merged model for each candidate. When the objective is a target-language benchmark it also needs labels in a language that by assumption has none; the surrogate readouts below remove that second cost but not the first. The constraint is usually checked after the fact, by measuring or reading the language of generated samples. Methods that tune coefficients through the merged model (Yang et al., 2024; Jung et al., 2024) also require model evaluations during search. The next section gives a way to score windows before building them. We use the first-response-token scalar because it is the differentiable readout needed by Theorem 1; this is an analytical compromise, not a claim that first-token confidence captures compounding autoregressive errors or mid-generation language collapse.

3 WHY GRAFTS ADD UP

The structure of (2) is cheaper than its values. Two per-layer quantities, both taken from the host's own forward and backward pass on a probe p, carry all of it. The *injection* v_ℓ is the change in what layer ℓ adds to the residual stream when the donor's layer replaces the host's, evaluated at the host's input to that layer. The *adjoint* s_ℓ^φ is the gradient of the readout with respect to the residual stream just after layer ℓ, with the later layers held at the host's:

$$v_\ell := B_\ell\big(x_\ell^{\mathrm{h}};\theta_\ell^{\mathrm{d}}\big) - B_\ell\big(x_\ell^{\mathrm{h}};\theta_\ell^{\mathrm{h}}\big), \qquad s_\ell^\varphi := \frac{\partial\varphi\big(x_L^{\mathrm{h}}\big)}{\partial x_{\ell+1}}, \qquad a_\ell^\varphi := \mathbb{E}_p\big\langle s_\ell^\varphi(p),\, v_\ell(p)\big\rangle. \tag{3}$$

Both v_ℓ and s_ℓ^φ live in $\mathbb{R}^{T\times d}$, and the inner product runs over all positions and dimensions, so a_ℓ^φ is in the units of the readout. All L injections come from evaluating each donor layer beside the host during one forward pass, and all L adjoints from one backward pass. The score a_ℓ^φ is the first-order change in the readout when layer ℓ alone is swapped; the result below says that these single-layer scores also predict every multi-layer graft.

Theorem 1 (Grafts add up)**.** *Suppose the layers and the readout are smooth near the host's trajectory (Assumption 1 in Appendix A). Then every graft set S that is admissible, meaning small enough for its trajectory to stay in that region, satisfies*

$$\Delta\varphi(S) = \sum_{\ell\in S} a_\ell^\varphi + r(S), \qquad |r(S)| \le C_1\,\lambda\,\varepsilon_S + C_2\,\varepsilon_S^2, \tag{4}$$

where $\varepsilon_S = \sup_p \sum_{\ell\in S} \|v_\ell(p)\|$ is the size of the graft, λ bounds the difference between the two experts' layer Jacobians along the host's trajectory, and C_1, C_2 depend only on the depth, the smoothness constants, and the readout.

Why it holds. Each injection is computed on the host's own trajectory, so it does not depend on which other layers are grafted, and the host carries each to the readout linearly to first order, so the effects add. The remainder collects what this misses: the curvature of each layer and of the readout, and the difference between how the two experts respond to the deviation earlier grafts already caused. The argument is a standard perturbation expansion; what matters is the expansion point, which makes every term independent of the rest of the graft. If both experts are fine-tuned by a small amount t from the same base, and the layers are smooth in their parameters, then λ and ε_S are $O(t)$, so the scores are $O(t)$ and the remainder $O(t^2)$, which Figure 2(b) tests. The constants grow exponentially with depth, so we never use them numerically.

The identity is what makes the program cheap. Every graft set is scored by a subset sum of L numbers per readout, so both the objective and the constraint of (2) are linear in the indicator of S, and one host pass per probe scores all 2^L compositions.

Scores rank candidates; they prove nothing about the one we pick. Two elementary facts close that gap, and Section 4 puts them to work: a score gap wider than the two remainders settles a comparison, and the largest remainder among a few built competitors bounds that of an unbuilt one drawn uniformly (Proposition 3, proved in Appendix A).

4 SUTURE: FROM SCORES TO A CERTIFIED CHOICE

SUTURE turns Section 3 into four steps (Algorithm 1): score every layer, calibrate the language scores on ten built windows, choose a band, and build it to certify the choice. Units are whole decoder layers. The composed model θ^S takes the donor's complete decoder layers at the indices in S, including their normalization weights, and takes everything else from the host, including the embedding, the final normalization, and the output head.

Readouts. Both readouts are next-token log-probability functionals at the first response position of a probe prompt p, so each costs one forward and one backward pass and needs no generation:

$$\varphi_{\mathsf{task}}(p) = \log \sum_{y\in\mathcal{T}_{\mathrm{ans}}(p)} P(y \mid p), \qquad \varphi_{\mathsf{lang}}(p) = \log \sum_{y\in\mathcal{T}_{\mathrm{own}}} P(y \mid p) - \log \sum_{y\in\mathcal{T}_{\mathrm{en}}} P(y \mid p). \tag{5}$$

A task probe is a question machine-translated from the *training* split of GSM8K (Cobbe et al., 2021) followed by "Answer:", with $\mathcal{T}_{\mathrm{ans}}(p)$ the tokens of its gold numeric answer, inherited from the English source. A language probe is a target-language passage followed by an instruction to respond in that language, with $\mathcal{T}_{\mathrm{own}}$ the tokens of the language's own name and $\mathcal{T}_{\mathrm{en}}$ those of "English". Both pass through the host's final normalization and output head, so they are differentiable in x_L, as Theorem 1 requires. For a multi-token answer or language name, these are deliberately bags of protocol-listed candidate tokens at the first response position, not sequence likelihoods: the adjoint only sees the distribution at that position. This choice keeps the scalar differentiable and makes clear what the proxy does not measure. How closely φ_{lang} agrees with the language of free generation has not been measured (Section 8); Appendix D gives templates, token sets, and probe counts.

Scores. For each readout, a_ℓ^φ is the probe average of $\langle s_\ell^\varphi, v_\ell \rangle$, as in (3), with the inner product over all positions and hidden dimensions. The injection v_ℓ is the output of donor layer ℓ minus that of host layer ℓ, both applied to the host's input to layer ℓ with the host's attention mask and position embeddings; s_ℓ^φ is the gradient of φ with respect to the output of host layer ℓ. Probes are scored one at a time, so no padding enters the scores.

Calibrate, then choose. A band is a window $[i, j]$ of consecutive layers, and $A_\varphi(S) = \sum_{\ell \in S} a_\ell^\varphi$. Scores rank windows well but can misstate magnitudes once a readout saturates, and the constraint depends on magnitudes (Section 5). SUTURE therefore builds $m_c = 10$ uniformly drawn windows, measures $\Delta\varphi_{\mathsf{lang}}$ on them, and fits the slope b of $\Delta\varphi_{\mathsf{lang}}$ on A_{lang} through the origin. It then selects

$$\widehat{S} = \arg\max_{S \in \mathcal{F}} A_{\mathsf{task}}(S), \qquad \mathcal{F} = \{\text{windows } S : b\, A_{\mathsf{lang}}(S) \geq -\tau\}, \tag{6}$$

the calibrated first-order version of (2). All $L(L+1)/2$ windows, 666 for a 36-layer model, have proxy scores computed exactly from prefix sums. We fix $\tau = 0.05$ nats before any measurement; we leave A_{task} uncalibrated because only its ordering enters this argmax; using the same builds to calibrate it could instead reduce the task-remainder variance in the certificate, but is not part of the default procedure. In the seeded stack the choice is stable across $\tau \in \{0.02, 0.05, 0.10, 0.20, 0.40\}$, selecting $[8, 10]$ at every value but the tightest, where it falls back to $[1, 2]$, and satisfying the true constraint in all five cases (Table 2). The implementation returns the best feasible non-empty window, then builds it and measures its true language change. If that change violates $-\tau$, it falls back to the best truly feasible calibration window, or to the host if none exists, and reports the violation instead of silently returning an infeasible choice. The host should also be kept if $\mathcal{F}$ is empty or that window's predicted gain is not positive. The conformal certificate requires $|\mathcal{F}| \geq m + 2$; when fewer feasible competitors exist, the selected window is returned without a certificate. The original 8B selections of Section 6 used $b = 1$; the isolated follow-up then fitted b and left those windows unchanged.

Build once and certify. Now $\theta^{\widehat{S}}$ is built and its language loss measured, together with m windows drawn uniformly without replacement from $\mathcal{F} \setminus \{\widehat{S}\}$ when $|\mathcal{F}| \geq m+2$. Measuring φ_{task} on each built window, on the probes used for scoring, gives its remainder $r(S) = \Delta\varphi_{\mathsf{task}}(S) - A_{\mathsf{task}}(S)$. With q the largest $|r|$ among the m draws, a competitor S is certified when $A_{\mathsf{task}}(\widehat{S}) - A_{\mathsf{task}}(S) > q + |r(\widehat{S})|$. By Proposition 3 an unbuilt competitor drawn uniformly has $|r(S)| \leq q$ with probability at least $m/(m+1)$, and a built one has $|r(S)| \leq q$ outright; either way it cannot then beat $\widehat{S}$. We use $m = 10$, bounding that failure probability by $1/11$ and bringing the procedure to 21 built models. The statement is about φ_{task} on the scoring probes, not about downstream accuracy, and it has been evaluated in controlled stacks but not on language models.

Donor layers are streamed one at a time during scoring, so peak memory is one host model plus one layer.

Algorithm 1: SUTURE. Lines 2–7 score every layer from one host forward–backward pass per probe; s_ℓ^φ and v_ℓ must come from the same host pass and probe. Lines 8–9 calibrate the language scores on ten built windows and choose; later lines build the choice, fall back if the measured language change violates the constraint, and otherwise certify it.

1: **Input:** host θ^{h}, donor θ^{d}, probes $\mathcal{P}_{\mathsf{task}}, \mathcal{P}_{\mathsf{lang}}$, tolerance τ, sizes m_c and m
2: **for** $\varphi \in \{\varphi_{\mathsf{task}}, \varphi_{\mathsf{lang}}\}$ and each $p \in \mathcal{P}_\varphi$ **do**
3: host forward pass on p, caching the input x_ℓ^{h} and the output of every layer
4: for each ℓ, apply donor layer ℓ to x_ℓ^{h} with the host's mask and positions; store v_ℓ {streamed}
5: backward pass from $\varphi(p)$, reading off s_ℓ^φ at every layer output
6: accumulate $a_\ell^\varphi \mathrel{+}= \langle s_\ell^\varphi, v_\ell \rangle / |\mathcal{P}_\varphi|$
7: **end for**
8: build m_c uniformly drawn windows; measure $\Delta\varphi_{\mathsf{task}}$ and $\Delta\varphi_{\mathsf{lang}}$; fit b from the latter against A_{lang} and retain them as $\mathcal{C}$
9: score every window by prefix sums; $\widehat{S} \leftarrow$ (6)
10: build $\widehat{S}$; measure its $\Delta\varphi_{\mathsf{task}}$ and $\Delta\varphi_{\mathsf{lang}}$
11: **if** $\Delta\varphi_{\mathsf{lang}}(\widehat{S}) < -\tau$ **then**
12: $\mathcal{C}_{\mathrm{feas}} \leftarrow \{S \in \mathcal{C} : \Delta\varphi_{\mathsf{lang}}(S) \geq -\tau\} \cup \{\emptyset\}$; set both host changes to 0
13: **return** the best $\widehat{S}_{\mathrm{fb}} \in \mathcal{C}_{\mathrm{feas}}$ by measured $\Delta\varphi_{\mathsf{task}}$, its measured language change, and the violation of $\widehat{S}$
14: **else**
15: **if** $|\mathcal{F}| \geq m + 2$ **then**
16: build m windows from $\mathcal{F} \setminus \{\widehat{S}\}$; measure each $r = \Delta\varphi_{\mathsf{task}} - A_{\mathsf{task}}$
17: $q \leftarrow$ the largest $|r|$ among the m draws
18: **return** $\widehat{S}$, its measured $\Delta\varphi_{\mathsf{lang}}$, and every $S \in \mathcal{F} \setminus \{\widehat{S}\}$ with $A_{\mathsf{task}}(\widehat{S}) - A_{\mathsf{task}}(S) > q + |r(\widehat{S})|$
19: **else**
20: **return** $\widehat{S}$ and its measured $\Delta\varphi_{\mathsf{lang}}$ without a certificate
21: **end if**
22: **end if**

5 CONTROLLED STACKS: THE IDENTITY, THE MECHANISM, AND ITS LIMIT

The claims of Sections 3 and 4 can be checked where exhaustive search is affordable: residual stacks with $L = 12$ blocks, $d = 24$ and $T = 5$ positions, two experts made from a shared base, and a next-token log-probability readout of the same form as the language-model readouts. Each block layer-normalizes its input, mixes positions causally, and applies a position-wise GELU MLP, so the trajectory carries the normalization and nonlinearity that the theorem's smoothness assumption has to survive; the mixing matrix is fixed rather than input-dependent, so these stacks do not test attention. Five questions follow the story. Does the identity hold? Does the score need both factors, and does it beat the closest baselines at an equal number of built models? Does selection respect a constraint? Is the certificate sound? Where does ranking stop? Two settings answer them: experts made by random perturbation, and experts obtained by gradient fine-tuning, scored and judged on disjoint inputs. The seeded-stack checks are recomputed by the retained NumPy harnesses. The multi-stack plots and tables use frozen aggregates. This is not language-model evidence, since here the ground truth *is* the readout the score is built from.

The identity holds. Predicted and measured change agree over random graft sets of every size, at Pearson 0.980 in the seeded stack; Figure 2(a) shows the per-stack Pearson values, with a multi-stack median of 0.95 over 48 stacks. The seeded-stack remainder falls with the fine-tuning scale at a fitted log-log slope of 2.055 (Figure 2(b)), while the aggregate over the three smallest scales in Table 2 gives 2.045.

The score needs both factors. Table 1 removes one factor at a time from $\langle s_\ell, v_\ell \rangle$. Windows have fixed width, so a score summing non-negative magnitudes cannot win by taking the largest one, and per-layer expert gaps span $4\times$, so parameter statistics have variation to rank. Read the regret column: unlike a rank correlation it does not depend on a sign an unsigned score cannot know. The full score selects at regret 0.006. Dropping the adjoint gives 0.21, dropping the injection 0.12, movement from the base 0.19, and random scores 0.20; every alternative is worse on the same stacks (Holm-adjusted $p \leq 0.016$), and 48 stacks leave the order unchanged (0.011 against 0.16–0.24, all $p < 10^{-5}$). Movement from the base, the statistic behind the original band (Bandarkar et al., 2025),

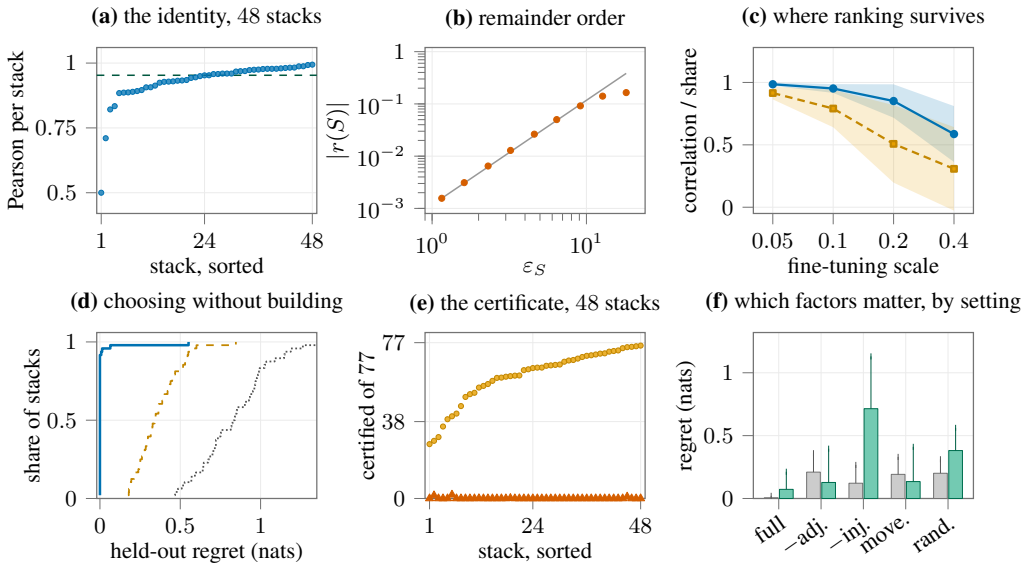

Figure 2: Controlled-stack checks. (a) Per-stack Pearson correlation; the dashed line is the median. (b) Seeded-stack remainder and a quadratic reference (grey). (c) Spearman correlation (blue) and certified share (orange); shading shows ± 1 standard deviation. (d) Held-out regret distributions for SUTURE (blue), random search (orange), and hill climbing (grey). (e) Certified competitors (orange) and false certifications (red triangles). (f) Width-6 regret for perturbed (grey) and trained (green) experts, with one-sided standard-deviation bars. Regret is in nats.

Table 1: Controlled-stack ranking and selection over 12 stacks (mean $\pm$ sd). Regret is the gap to the best window, in nats. Builds count merged models used before selection. Differences are paired against SUTURE, with bootstrap 95% intervals and Holm-adjusted Wilcoxon tests.

	Builds	Spearman	Regret	Δregret [95% CI]	p_{Holm}
Seven windows of width 6, 12 stacks					
SUTURE, $\langle s_\ell, v_\ell \rangle$	0	0.93 ± 0.10	0.006 ± 0.015		
drop the adjoint, $\|v_\ell\|$	0	-0.07 ± 0.53	0.210 ± 0.151	$+0.204$ [0.125, 0.282]	0.008
drop the injection, $\|s_\ell\|$	0	0.19 ± 0.68	0.121 ± 0.145	$+0.116$ [0.043, 0.202]	0.016
movement from base, $\|\Delta\theta_\ell^{\mathrm{d}}\|$	0	0.06 ± 0.63	0.192 ± 0.139	$+0.186$ [0.113, 0.256]	0.008
random scores	0	0.00 ± 0.05	0.201 ± 0.111	$+0.195$ [0.142, 0.253]	0.002
All 78 windows, 12 stacks					
SUTURE	0	0.95 ± 0.04	0.013 ± 0.018		
first-order weight attribution	0	0.94 ± 0.03	0.017 ± 0.020	$+0.004$ [0.000, 0.011]	1.000
Sens-Merging-style sensitivity	0	0.02 ± 0.44	0.241 ± 0.192	$+0.228$ [0.125, 0.337]	0.003
best of 3 random windows	3	*n/a*	0.154 ± 0.071	$+0.141$ [0.102, 0.186]	0.003
best of SUTURE's top 3	3	*n/a*	0.010 ± 0.019	-0.003 [-0.008, -0.0001]	0.125
best of 11 random windows	11	*n/a*	0.062 ± 0.033	$+0.049$ [0.028, 0.073]	0.006
Specificity control: score for readout A or B, ranked against B					
score for A	0	-0.04 ± 0.44	*n/a*		
score for B	0	0.92 ± 0.06	*n/a*		

is not distinguishable from random scores here (0.192 against 0.201, paired $p = 0.91$), because nothing in this construction ties parameter magnitude to the readout; width-3 windows give the same picture (Appendix B). The specificity rows are a control: the score built for readout B ranks B at 0.92 while the score for A does not (-0.04), matching the near-zero correlation between the two readouts' effects (-0.06).

Against the closest baselines. First-order weight attribution, the closest per-layer score, is indistinguishable from SUTURE here. It ranks worse only as fine-tuning grows: Spearman 0.49 against 0.59 at scale 0.4, with no significant regret difference (Table 3). A Sens-Merging-style sensitivity (Liu

et al., 2025) is no more distinguishable from random scores (0.241 against 0.264, paired $p = 0.42$) than movement was. Search pays in builds: the best of 11 random windows still has regret 0.062, against 0.013 for scores that build nothing.

Trained experts and held-out inputs. Random perturbations are a weak testbed, and scoring and judging on one input is a weak protocol. We repeat the comparison with gradient-trained experts, scored on eight probe inputs and judged on eight disjoint held-out ones, every build-based baseline choosing among its built windows by the same probe readout (Table 4). Held-out ranking stays high over all 78 windows (0.98 ± 0.02 over 48 stacks). Table 4 reports the seven width-6 diagnostic, where SUTURE reaches 0.64 ± 0.48 and still beats random scores and both one-factor ablations, but gradient training couples movement from the base to the injections (Pearson 0.87), so movement selects almost as well and neither it nor weight attribution separates from the full score. A readout-agnostic statistic can stand in here because the construction aligns the two behaviors, the truths anticorrelating at -0.96; under random perturbation they are unrelated (-0.06) and movement is no better than random (Appendix B). Which regime real pairs fall into is untested. What the scores buy is builds (Table 4; Figure 2(d)): with none, SUTURE reaches regret 0.013 against 0.002 for a full 78-window sweep, a difference that is not significant (Holm-adjusted $p = 0.95$), while random search and hill climbing with 11 builds stay far behind. Figure 2(f) summarizes the corresponding width-6 regret contrast between perturbed and trained experts.

Selection under a constraint needs calibration. The constraint is where the two settings part. Under random perturbation, with readout B as the constraint and thresholds on the true scale, the selected window is truly feasible in 0.94, 0.79 and 0.83 of draws. With trained experts the raw language scores understate the loss so badly that the selection is feasible in 4, 3 and 0 of 48 stacks. Fitting one slope on ten built windows repairs it, to 0.93, 0.85 and 0.79 of draws, which is why Section 4 calibrates first. Ranking was never the problem: the language scores rank the true loss at Spearman 0.98.

The certificate behaves as predicted. In the seeded stack the selection is the true optimum of all 78 windows, and the margin condition certifies 75 competitors with every remainder known, 71 from $m = 10$ sampled ones, neither falsely. Over 12 stacks it certifies 61.3 ± 11.1 of 77 from 11 built models with none false; over 48 stacks it makes 0.09 false certifications per draw, 0.15% of all certifications, consistent with a guarantee that holds per competitor rather than simultaneously (Figure 2(e)). Growing m from 5 to 40 lowers that rate to 0.02 at a small cost in certified competitors (Table 3).

Where ranking stops. Figure 2(c) shows the multi-stack diagnostic: mean ranking and certification deteriorate as the experts move away from the base, reaching Spearman 0.59 and a certified share of 0.31 at scale 0.4. Depth does the same, to 0.89 at depth 24, and false certifications become frequent in both regimes (Table 3). Magnitude error stays worse than order: 9% and 18% at scales 0.05 and 0.1, where the order is exact (Table 2). In the worst of 48 stacks (superposition 0.50) the selection was still optimal and the certificate shrank to about 30 of 77 competitors, none false. SUTURE ranks; it does not predict how much a composition will help, and it should not be trusted beyond the scale at which ranking breaks.

6 LANGUAGE MODELS: WHAT HOLDS AT SCALE, AND WHAT DOES NOT YET

Controlled stacks cannot show that the ranking survives on real experts or under free generation. We report three real-model studies, each under a protocol fixed before its measurement (Appendix D).

Public 8B specialists. We took the Qwen3-8B native specialists of Lasbordes et al. (2026) for French and Chinese as hosts and their English specialist as donor. Selection scored all 666 windows of the 36-layer model, with no grafted model built, and chose layers $[13, 15]$ for French and $[20, 31]$ for Chinese (zero-based). On 250 MGSM-Rev2 test records per language, disjoint from the probes, the selected windows raised mean teacher-forced answer log-probability over the host by $+0.398$ nats per token in French (paired bootstrap 95% interval $[+0.358, +0.437]$) and by $+1.326$ in Chinese ($[+1.210, +1.446]$). The host means were -8.808 and -9.516 nats per token, respectively. The

same records were used to measure the published windows (layers 13–22 and 13–20): selected minus published was -0.355 nats per token in French (paired bootstrap 95% interval $[-0.417, -0.293]$) and -0.131 in Chinese ($[-0.222, -0.039]$), so the selected intervals beat the host but lose to the published references. All intervals use zero-based inclusive layer indices: the published $[13, 22]$ and $[13, 20]$ intervals therefore match the selected intervals' convention. Three limits matter: the answer readout is close to but not the task readout, and neither is free-generation accuracy; a decoding-cache mismatch prevented the latter. An isolated A100 follow-up fit the language scale on ten windows: $b = 0.452$ (RMSE 0.747) for French and $b = 0.012$ (RMSE 0.916) for Chinese, leaving the selected windows $[13, 15]$ and $[20, 31]$. Parameter movement selected $[0, 35]$ in both languages, while first-order weight attribution selected the same constrained windows; without the constraint it selected French $[22, 32]$, gaining $+1.243$ answer nats but losing -1.259 language-readout units. The conformal certificate was not run. Algorithm 1 specifies a measured calibration-window fallback if a future choice violates the constraint. The Chinese window extends past layer 20, where Lasbordes et al. (2026) saw traces revert to English. On 76 held-out language probes, selected-minus-host was $(+1.070, +0.268)$ and selected-minus-published $(+0.130, -0.315)$ (French, Chinese). The proxy supports a host-relative gain but not the Chinese comparison with the published window; generated-text language identification remains unmeasured.

A weak 1.5B pair: free-generation ranking is inconclusive. On Qwen2.5-Coder-1.5B-Instruct, the only 1–2B decoder in our offline mirror, matched LoRA experts gave a donor that added one correct item of 32 in Spanish and Chinese and none in Swahili. Over all 406 Spanish windows the scores ranked teacher-forced log-probability on eight MGSM development items (Shi et al., 2022) at Spearman 0.877, but free-generation exact match at -0.018. With 40 Bernoulli trials per window near a rate of 0.1, that value is indistinguishable from chance (Appendix D), and a later greedy ranking on 64 held-out items reached only 0.129, below the pre-specified 0.5. This donor was trained on the GSM8K questions the probes are drawn from, so its probe scores are in-sample, while the ranking targets come from the disjoint MGSM split. On 24 fixed windows an adjoint-only score ranked teacher-forced outcomes better than the full score (0.803 against 0.583), the reverse of the controlled ablation. A third attempt, on Qwen3-1.7B, stopped before ranking at the host-language and implementation checks (Appendix D).

7 Related work

Three literatures meet here: layer swapping, model-merging search, and first-order attribution.

Layer swapping and multilingual composition. Bandarkar et al. (2025) introduced layer swapping and selected its band from expert parameter movement; later work found little overlap between math- and language-important parameters (Bandarkar & Peng, 2025), extended the recipe to long reasoning and safety (Lasbordes et al., 2026; Shin & Hwang, 2026), and made merging input-steerable (Li et al., 2026). This lineage includes task arithmetic and composable sparse fine-tuning (Ilharco et al., 2023; Chronopoulou et al., 2024; Ansell et al., 2021), but these methods do not predict a band's behavioral effect before construction; Theorem 1 does so to first order.

How compositions are chosen. Evolutionary, Bayesian, and surrogate methods still evaluate individual merges (Akiba et al., 2025; Mencattini et al., 2025; Chen et al., 2026; Akizuki et al., 2025); expert-intrinsic rules read weights rather than behavior (Yadav et al., 2023; Yu et al., 2024; Jin et al., 2023; Liu et al., 2025); and behavior-referenced or unlabeled proxies score one configuration at a time (Yang et al., 2024; Jung et al., 2024; Fu et al., 2026; de Jorge et al., 2026). StatsMerging (Merugu et al., 2025) and CAT Merging (Sun et al., 2025) likewise operate in weight space, while MergeBench (He et al., 2025) benchmarks merged LLMs; none scores combinatorial layer grafts under a language constraint.

Attribution and output language. Attribution patching and Weight Patching linearize activation or parameter changes (Nanda, 2023; Syed et al., 2024; Sun et al., 2026); we do not claim that family as new (Table 1). Our distinction is exact host-path differencing for two independently tuned experts, a constrained program over all 2^L sets, and a per-competitor remainder certificate. Interpretability studies place language-agnostic computation mid-stack and language commitment late (Wendler

et al., 2024; Tang et al., 2024; Schut et al., 2025); work on confusion and late-layer failures (Marchisio et al., 2024; Nie et al., 2025) motivates retaining a language readout.

8 Implications and Limitations

What a practitioner gets. A fixed budget of twenty-one builds calibrates and certifies one choice rather than searching; a published band costs one build but is tied to its model and language, and can send Chinese traces back to English (Lasbordes et al., 2026). The same scores rank arbitrary subsets at no extra cost, but that extension is untested on language models.

Why swaps compose, and where they should stop. When the language readout tracks output language, the theorem predicts a stopping boundary. Our held-out proxy supports only a host-relative teacher-forced gain, not the Chinese selected-versus-published comparison; Appendix C shows why the free-generation claim may fail.

Limitations. At 8B the evidence covers two languages, one donor, and a teacher-forced functional; selected intervals lose to published references, and generated-text language identification is unmeasured. The certificate remains controlled-stack only; an isolated follow-up ran 8B calibration and cheaper baselines. The 1.5B study is inconclusive; local studies use one seed and machine-translated probes. Fixed-attention stacks achieve only 0.64 ± 0.48 on the width-6 diagnostic; a from-scratch transformer (Appendix C) also failed to reproduce transfer.

Reproducibility Statement

Assumptions and proofs are in Appendix A. The released NumPy/SciPy code recomputes the seeded controlled-stack checks and their figure data. Per-stack records for the 12- and 48-stack studies are released under results/controlled_stacks/. verify/audit_controlled_results.py recomputes their means, paired tests, bootstrap intervals, and plotted values. The original multi-stack generator and per-candidate window scores are absent, so a full rerun from initialization is not available. The released 8B answer measurements contain per-record values and manifests; verify/audit_b3_results.py checks their means and paired intervals. The 76-probe language-fidelity measurements are also released with their translated probe inputs, and the same script recomputes their paired intervals. Other isolated follow-ups have only aggregate summaries, so their intervals cannot be recomputed from this release. Appendix D lists checkpoints, data, and software. Weights are not redistributed. Code and aggregate evidence are available at https://anonymous.4open.science/r/grafts-add-up-8541/.

Ethics Statement

This work uses existing open models and no human subjects or new data collection. Layer grafting can transfer unsafe behavior or weaken donor-language safety tuning; it is not a safety guarantee. Machine-translated probes may carry uneven artifacts. The intended benefit is non-English capability without in-language annotation.

AI Use Statement

Generative AI assisted the theory, proofs, hypotheses, protocols, code, manuscript, figures, literature search, and reference checks. It also translated probes with the pinned Qwen models in Appendix D; NumPy controlled-stack data were AI-assisted, but AI supplied neither labels nor reported numbers. The authors re-derived and checked the theorems, optimizers, scripts, artifacts, and bibliography. No run used a hosted API: the 1.5B/1.7B runs used a local GPU and the 8B run a rented A100 with open weights. We take responsibility for the final content.

References

Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. Evolutionary optimization of model merging recipes. *Nature Machine Intelligence*, 7(2):195–204, 2025. doi: 10.1038/s42256-024-00975-8.

Rio Akizuki, Yuya Kudo, Nozomu Yoshinari, Yoichi Hirose, Toshiyuki Nishimoto, Kento Uchida, and Shinichi Shirakawa. Surrogate benchmarks for model merging optimization. In *International Conference on Automated Machine Learning (AutoML), Non-Archival Content Track*, 2025. doi: 10.48550/arXiv.2509.02555. arXiv:2509.02555.

Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023. doi: 10.1561/2200000101.

Alan Ansell, Edoardo Maria Ponti, Anna Korhonen, and Ivan Vulić. Composable sparse fine-tuning for cross-lingual transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1778–1796, 2021. doi: 10.18653/v1/2022.acl-long.125.

Lucas Bandarkar and Nanyun Peng. The unreasonable effectiveness of model merging for cross-lingual transfer in LLMs. In *Proceedings of the 5th Workshop on Multilingual Representation Learning (MRL)*, pp. 131–148, 2025. doi: 10.18653/v1/2025.mrl-main.10.

Lucas Bandarkar, Benjamin Muller, Pritish Yuvraj, Rui Hou, Nayan Singhal, Hongjiang Lv, and Bing Liu. Layer swapping for zero-shot cross-lingual transfer in large language models. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. doi: 10.48550/arXiv.2410.01335. Spotlight. arXiv:2410.01335.

Kesheng Chen, Yamin Hu, Zhenqian Zhu, Yiya Diao, and Wenjian Luo. AP-BMM: Approximating capability-cost pareto sets of LLMs via asynchronous prior-guided Bayesian model merging. *arXiv preprint arXiv:2512.09972*, 2026. doi: 10.48550/arXiv.2512.09972. URL https://arxiv.org/abs/2512.09972.

Alexandra Chronopoulou, Jonas Pfeiffer, Joshua Maynez, Xinyi Wang, Sebastian Ruder, and Priyanka Agrawal. Language and task arithmetic with parameter-efficient layers for zero-shot summarization. In *Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL)*, pp. 114–126, 2024. doi: 10.18653/v1/2024.mrl-1.7.

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934. doi: 10.1093/BIOMET/26.4.404.

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

Pau de Jorge, César Roberto de Souza, Björn Michele, Mert Bülent Sarıyıldız, Philippe Weinzaepfel, Florent Perronnin, Diane Larlus, and Yannis Kalantidis. Task alignment: A simple proxy for practical model merging across diverse vision tasks. *arXiv preprint arXiv:2604.12935*, 2026. doi: 10.1007/978-3-032-37098-3_25. Accepted at ECCV 2026.

Zihang Fu, Haonan Wang, Jian Kang, Kenji Kawaguchi, and Jiaying Wu. Lost in adaptation: Layer-selective recovery of temporal reasoning in video-language models. *arXiv preprint arXiv:2604.11399*, 2026. doi: 10.48550/arXiv.2604.11399. URL https://arxiv.org/abs/2604.11399.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2018. doi: 10.63317/2dwkd5xar3i5.

Yifei He, Siqi Zeng, Yuzheng Hu, Rui Yang, Tong Zhang, and Han Zhao. MergeBench: A benchmark for merging domain-specialized LLMs. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2025. doi: 10.52202/085713-3374.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. doi: 10.48550/arXiv.2212.04089.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. doi: 10.48550/arXiv.2212.09849. arXiv:2212.09849.

Aecheon Jung, Seunghwan Lee, Dongyoon Han, and Sungeun Hong. SyMerge: From non-interference to synergistic merging via single-layer adaptation. *arXiv preprint arXiv:2412.19098*, 2024.

Maxence Lasbordes, Amélie Chatelain, and Djamé Seddah. Rethinking the multilingual reasoning gap with layer swap. *arXiv preprint arXiv:2605.26735*, 2026. doi: 10.48550/arXiv.2605.26735.

Zhuoran Li, Rui Xu, Jian Yang, Junnan Liu, Zhijun Chen, Qianren Mao, Hongcheng Guo, Jiaheng Liu, Likang Xiao, Ming Li, and Xiaojie Wang. Enhancing multilingual reasoning via steerable model merging. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 37266–37277, 2026. doi: 10.18653/v1/2026.findings-acl.1856.

Shuqi Liu, Han Wu, Bowei He, Xiongwei Han, Mingxuan Yuan, and Linqi Song. Sens-Merging: Sensitivity-guided parameter balancing for merging large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19243–19255, 2025. doi: 10.18653/v1/2025.findings-acl.984.

Kelly Marchisio, Wei-Yin Ko, Alexandre Berard, Théo Dehaze, and Sebastian Ruder. Understanding and mitigating language confusion in LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6653–6677, 2024. doi: 10.18653/v1/2024.emnlp-main.380.

Tommaso Mencattini, Robert Adrian Minut, Donato Crisostomi, Andrea Santilli, and Emanuele Rodolà. MERGE^3: Efficient evolutionary merging on consumer-grade GPUs. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 43694–43715, 2025. doi: 10.48550/arXiv.2502.10436. PMLR volume 267; arXiv:2502.10436.

Ranjith Merugu, Bryan Bo Cao, and Shubham Jain. StatsMerging: Statistics-guided model merging via task-specific teacher distillation. *arXiv preprint arXiv:2506.04567*, 2025. doi: 10.48550/arXiv.2506.04567.

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

Ercong Nie, Helmut Schmid, and Hinrich Schütze. Mechanistic understanding and mitigation of language confusion in English-centric large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025. doi: 10.18653/v1/2025.findings-emnlp.37. ACL Anthology 2025.findings-emnlp.37; arXiv:2505.16538.

Lisa Schut, Yarin Gal, and Sebastian Farquhar. Do multilingual LLMs think in English? *arXiv preprint arXiv:2502.15603*, 2025. doi: 10.48550/arXiv.2502.15603.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*, 2022. doi: 10.48550/arXiv.2210.03057.

Hyunseo Shin and Wonseok Hwang. Layer-wise swapping for generalizable multilingual safety. In *Proceedings of the 2026 Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 2223–2238, 2026. doi: 10.18653/v1/2026.eacl-long.98. arXiv:2601.22620.

Chenghao Sun, Chengsheng Zhang, Guanzheng Qin, Rui Dai, and Xinmei Tian. Weight patching: Toward source-level mechanistic localization in LLMs. *arXiv preprint arXiv:2604.13694*, 2026. doi: 10.48550/arXiv.2604.13694.

Wenju Sun, Qingyong Li, Yangliao Geng, and Boyang Li. CAT merging: A training-free approach for resolving conflicts in model merging. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57523–57543. PMLR, 2025. doi: 10.48550/arXiv.2505.06977.

Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, 2024. doi: 10.18653/v1/2024.blackboxnlp-1.25.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. Language-specific neurons: The key to multilingual capabilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 5701–5715, 2024. doi: 10.18653/v1/2024.acl-long.309.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005. doi: 10.1007/b106715.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Do Llamas work in English? on the latent language of multilingual transformers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 15366–15394, 2024. doi: 10.18653/v1/2024.acl-long.820.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. doi: 10.52202/075280-0310.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. AdaMerging: Adaptive model merging for multi-task learning. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. doi: 10.48550/arXiv.2310.02575.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *International Conference on Machine Learning (ICML)*, 2024. doi: 10.48550/arXiv.2311.03099.

A THEORY: ASSUMPTION AND PROOFS

This appendix states the assumption behind Theorem 1, proves the theorem through an identity on the residual stream, and proves Proposition 3. Norms are Frobenius norms on $\mathbb{R}^{T\times d}$ and the induced operator norms on linear maps of that space, and $\langle\cdot,\cdot\rangle$ is the matching inner product. Each layer map $B_\ell(\cdot\,;\theta)$ includes the probe's fixed attention mask and position embeddings. We fix one probe until the last step of the proof of Theorem 1, write $\Delta_\ell = x_\ell^S - x_\ell^{\mathrm{h}}$, with $\Delta_0 = 0$ because the embedding is the host's, and write $a(\ell) = \mathrm{d}$ if $\ell \in S$ and $a(\ell) = \mathrm{h}$ otherwise.

Assumption 1 (Smoothness near the host trajectory)**.** There are constants $\rho, \Lambda, \kappa, \lambda, L_\varphi \geq 0$ such that, for every probe and every layer ℓ, on the ball $\mathcal{B}_\ell$ of radius ρ around the host state x_ℓ^{h}: the map $x \mapsto B_\ell(x;\theta)$ is Λ-Lipschitz with a κ-Lipschitz Jacobian for θ equal to the host's or the donor's layer parameters; and $\|J_\ell^{\mathrm{d}} - J_\ell^{\mathrm{h}}\| \leq \lambda$, where $J_\ell^\bullet$ is the Jacobian of layer ℓ at x_ℓ^{h} under parameters $\bullet$. The readout has an L_φ-Lipschitz gradient on the ball of radius ρ around x_L^{h}. A graft set S is *admissible* if $\gamma\,\varepsilon_S \leq \rho$, where $\gamma = (1+\Lambda)^L$.

Because Λ may depend on ρ, the two are chosen together. Nothing here constrains smoothness in the parameters; only the scaling remark below uses it.

Lemma 2 (Superposition in the residual stream)**.** *Let $P_{\ell+1:L} = \prod_{k=L-1}^{\ell+1}\big(I + J_k^{\mathrm{h}}\big)$, the Jacobian of the host map from the residual stream after layer ℓ to x_L (the empty product is I), and let $\Gamma = \max_\ell \|P_{\ell+1:L}\|$. Under Assumption 1, every admissible S satisfies*

$$\Delta_L = \sum_{\ell\in S} P_{\ell+1:L}\,v_\ell + R(S), \qquad \|R(S)\| \leq \Gamma\,\gamma\,\varepsilon_S\Big(|S|\,\lambda + \tfrac{\kappa}{2}\,L\,\gamma\,\varepsilon_S\Big),$$

and $\Gamma \leq \gamma$.

Proof of Lemma 2. By definition of the two trajectories,

$$\Delta_{\ell+1} = \Delta_\ell + \underbrace{\Big[B_\ell\Big(x_\ell^{\mathrm{h}};\theta_\ell^{a(\ell)}\Big) - B_\ell\big(x_\ell^{\mathrm{h}};\theta_\ell^{\mathrm{h}}\big)\Big]}_{=\,\mathbf{1}[\ell\in S]\,v_\ell} + \underbrace{\Big[B_\ell\Big(x_\ell^{\mathrm{h}}+\Delta_\ell;\theta_\ell^{a(\ell)}\Big) - B_\ell\Big(x_\ell^{\mathrm{h}};\theta_\ell^{a(\ell)}\Big)\Big]}_{\text{propagation}}, \tag{7}$$

which is exact, since it only adds and subtracts $B_\ell(x_\ell^{\mathrm{h}};\theta_\ell^{a(\ell)})$; it is verified numerically to relative error 1.9×10^{-15} (Table 2), and the retained verify/verify_theory.py harness checks the recursion and the second-order decay of the readout remainder numerically; this is a numerical check, not a proof of the bound. If $\|\Delta_\ell\| \leq \rho$ the segment from x_ℓ^{h} to $x_\ell^{\mathrm{h}} + \Delta_\ell$ lies in the convex ball $\mathcal{B}_\ell$, so the propagation term has norm at most $\Lambda\|\Delta_\ell\|$ and $\|\Delta_{\ell+1}\| \leq (1+\Lambda)\|\Delta_\ell\| + \mathbf{1}[\ell\in S]\|v_\ell\|$. With $\Delta_0 = 0$, induction gives

$$\|\Delta_\ell\| \;\leq\; \sum_{k\in S,\,k<\ell} (1+\Lambda)^{\ell-1-k}\|v_k\| \;\leq\; \gamma\,\varepsilon_S \;\leq\; \rho \qquad (\ell = 0,\dots,L), \tag{8}$$

the last step being admissibility, which is what keeps the hypothesis $\|\Delta_\ell\| \leq \rho$ true at every step.

Because the Jacobian of $B_\ell(\cdot\,;\theta_\ell^{a(\ell)})$ is κ-Lipschitz on $\mathcal{B}_\ell$, the propagation term equals $J_\ell^{a(\ell)}\Delta_\ell + e_\ell$ with $\|e_\ell\| = \big\|\int_0^1 (J_\ell^{a(\ell)}(x_\ell^{\mathrm{h}} + u\Delta_\ell) - J_\ell^{a(\ell)})\Delta_\ell\,du\big\| \leq \frac{\kappa}{2}\|\Delta_\ell\|^2$. Writing $J_\ell^{a(\ell)} = J_\ell^{\mathrm{h}} + \mathbf{1}[\ell \in S](J_\ell^{\mathrm{d}} - J_\ell^{\mathrm{h}})$,

$$\Delta_{\ell+1} = \big(I + J_\ell^{\mathrm{h}}\big)\,\Delta_\ell + \mathbf{1}[\ell\in S]\,v_\ell + q_\ell, \qquad \|q_\ell\| \leq \mathbf{1}[\ell\in S]\,\lambda\|\Delta_\ell\| + \tfrac{\kappa}{2}\|\Delta_\ell\|^2. \tag{9}$$

Unrolling this linear recursion from $\Delta_0 = 0$ gives $\Delta_L = \sum_{\ell\in S} P_{\ell+1:L}v_\ell + \sum_\ell P_{\ell+1:L}q_\ell$, so $R(S) = \sum_\ell P_{\ell+1:L}q_\ell$. With (8), $\sum_\ell\|q_\ell\| \leq |S|\lambda\gamma\varepsilon_S + L\frac{\kappa}{2}\gamma^2\varepsilon_S^2$, and $\|P_{\ell+1:L}\| \leq \Gamma$ gives the bound. Finally a Λ-Lipschitz map has Jacobian norm at most Λ, so $\|I + J_k^{\mathrm{h}}\| \leq 1 + \Lambda$ and $\Gamma \leq (1+\Lambda)^L = \gamma$. □

Proof of Theorem 1. By (8) at $\ell = L$, the segment from x_L^{h} to x_L^S lies in the ball on which $\nabla\varphi$ is L_φ-Lipschitz, so $\varphi(x_L^S) - \varphi(x_L^{\mathrm{h}}) = \langle\nabla\varphi(x_L^{\mathrm{h}}),\Delta_L\rangle + \eta$ with $|\eta| \leq \frac{L_\varphi}{2}\|\Delta_L\|^2 \leq \frac{L_\varphi}{2}\gamma^2\varepsilon_S^2$. By the chain rule, $P_{\ell+1:L}^\top\nabla\varphi(x_L^{\mathrm{h}})$ is the gradient of the readout with respect to the residual stream after

layer ℓ, that is, s_ℓ^φ, so $\langle \nabla\varphi(x_L^{\mathrm{h}}), P_{\ell+1:L} v_\ell\rangle = \langle s_\ell^\varphi, v_\ell\rangle$. Substituting Lemma 2 and applying the Cauchy–Schwarz inequality to $\langle \nabla\varphi(x_L^{\mathrm{h}}), R(S)\rangle$ gives, for this probe,

$$\Big|\varphi(x_L^S) - \varphi(x_L^{\mathrm{h}}) - \sum_{\ell\in S}\langle s_\ell^\varphi, v_\ell\rangle\Big| \le G\,\Gamma\,\gamma\,|S|\,\lambda\,\varepsilon_S + \Big(\tfrac{\kappa}{2}\,G\,\Gamma\,L + \tfrac{L_\varphi}{2}\Big)\gamma^2\varepsilon_S^2,$$

where $G = \sup_p \|\nabla\varphi(x_L^{\mathrm{h}}(p))\|$. The right side does not depend on the probe, so averaging over probes proves the theorem with $C_1 = G\,\Gamma\,\gamma\,L \le G\,L\,\gamma^2$ and $C_2 = \big(\tfrac{\kappa}{2}G\,\Gamma\,L + \tfrac{L_\varphi}{2}\big)\gamma^2 \le \big(\tfrac{\kappa}{2}G\,L\,\gamma + \tfrac{L_\varphi}{2}\big)\gamma^2$, using $|S| \le L$ and $\Gamma \le \gamma$. The two constants carry different units, readout per state unit and readout per squared state unit, which is why the bound keeps them apart. □

Scaling with the fine-tuning size. Suppose, in addition, that B_ℓ and its Jacobian are Lipschitz in the parameters near the base, that $\theta^{\mathrm{h}} = \theta^{\mathrm{b}} + t\,u^{\mathrm{h}}$ and $\theta^{\mathrm{d}} = \theta^{\mathrm{b}} + t\,u^{\mathrm{d}}$, and that the constants of Assumption 1 can be taken uniform for small t. Then $\|v_\ell\| = O(t)$ and $\lambda = O(t)$, so the scores are $O(t)$ and $|r(S)| = O(t^2)$. This is the claim that Figure 2(b) tests.

Proposition 3 (Checking a choice)**.** *Let $A(S) = \sum_{\ell\in S} a_\ell^\varphi$, and let $\widehat{S}$ maximize A over a candidate set $\mathcal{F}$. **(i)** For any S, if $A(\widehat{S}) - A(S) > |r(S)| + |r(\widehat{S})|$, then $\Delta\varphi(\widehat{S}) > \Delta\varphi(S)$. **(ii)** If $|\mathcal{F}| \ge m+2$ and $S_1, \dots, S_{m+1}$ are drawn uniformly without replacement from $\mathcal{F}\setminus\{\widehat{S}\}$, then $\Pr\big[\,|r(S_{m+1})| > \max_{i\le m}|r(S_i)|\,\big] \le 1/(m+1)$. Part (ii) is the rank argument behind conformal prediction (Vovk et al., 2005; Angelopoulos & Bates, 2023).*

Proof of Proposition 3. (i) $\Delta\varphi(\widehat{S}) - \Delta\varphi(S) = A(\widehat{S}) - A(S) + r(\widehat{S}) - r(S) > |r(S)| + |r(\widehat{S})| + r(\widehat{S}) - r(S) \ge 0$. (ii) Once the probes are fixed, every $r(S)$ is a fixed number, so the only randomness is the draw, and $|\mathcal{F}| \ge m+2$ guarantees $m+1$ distinct competitors. Drawing without replacement makes $|r(S_1)|, \dots, |r(S_{m+1})|$ exchangeable, so the last of them is the strict maximum with probability at most $1/(m+1)$, with equality when ties have probability zero. For the certificate of Section 4, a competitor drawn uniformly from those not built forms such an exchangeable sequence with the m built ones. It can be falsely certified only if its $|r|$ exceeds q, since otherwise part (i) applies, so a false certification has probability at most $1/(m+1)$, and by linearity of expectation the expected number of false certifications among the $|\mathcal{F}| - m - 1$ unbuilt competitors is at most $(|\mathcal{F}| - m - 1)/(m+1)$. A built competitor has $|r| \le q$ by the definition of q and cannot be falsely certified. □

What the results do not say. Part (ii) concerns a uniformly drawn competitor, not an adversarial one. Theorem 1 does not say the remainder is small at any particular scale, and its constants are exponential in depth. It does not license reading $\sum_{\ell\in S} a_\ell^\varphi$ as an estimate of how much a graft helps: the order of the windows is exact at scales where the magnitude error is already 9% and 18% (Table 2). It assumes a shared architecture and base and differentiable layers, excluding hard-routed mixture-of-experts blocks without smoothing.

B Full verification table

Table 2 collects the consistency checks that back claims in the paper; the released harnesses print a longer list, and the supplement's readme says which script produces each entry.

Table 2: Full consistency table for the controlled stack, grouped as identities, machinery, and selection. Single-stack rows use the seeded stack of verify/verify_theory.py; the seeded-stack Pearson 0.980 comes from the separately seeded draw of verify/make_figdata.py. Multi-stack rows are frozen study aggregates with per-layer expert gaps spanning a $4\times$ range; the minimal public harness does not regenerate those aggregates. These checks corroborate the formal arguments; they are not evidence about language models.

Claim	Check	Result
Exact recursion (App. A)	rebuild Δ_L from the recursion	rel. err. 1.9×10^{-15}
Superposition, 12 stacks	fresh experts and readouts	Pearson 0.943 ± 0.034
Superposition, 48 stacks	same design	median 0.953, range 0.50–0.99
Remainder is second order (Thm. 1)	log-log slope, 3 smallest scales	2.045 *(bound: 2)*
Ranks but does not estimate (Thm. 1)	Spearman and relative magnitude error, seeded stack	order exact at scales 0.05 and 0.1; error 9% and 18%
Constraint respected across τ (Sec. 4)	five tolerances, seeded stack	5/5 selections satisfy the true constraint
Conformal clause (Prop. 3)	3000 draws per m from the competitors	0.096/0.046/0.021 vs. 0.091/0.048/0.024
Oracle margin, 12 stacks (Prop. 3)	true $\lvert r\rvert$ for every competitor	68.0 ± 9.1 of 77, none false
Deployable margin, 12 stacks (Prop. 3)	m=10 built, 200 draws per stack	61.3 ± 11.1 of 77, none false
Deployable margin, 48 stacks (Prop. 3)	same	60.5 ± 13.1 of 77; 0.09 false per draw
Selection, seeded stack	78 windows, predicted vs. built	regret 0.000, Spearman 0.979
Selection, 12 stacks	78 windows	regret 0.013 ± 0.018, Spearman 0.95 ± 0.04
Mechanism ablation, width 3, 12 stacks	10 windows, paired Wilcoxon, Holm	SUTURE 0.002 vs. 0.15–0.23; $p \leq 0.004$
Mechanism ablation, width 6, 48 stacks	7 windows, paired Wilcoxon, Holm	SUTURE 0.011 vs. 0.16–0.24; $p < 10^{-5}$
Expert-gap heterogeneity	c.v. of $\lVert\theta_\ell^{\mathrm{d}} - \theta_\ell^{\mathrm{h}}\rVert$	0.39 (0.019 with equal per-layer scales)

The tables below report the controlled experiments. The retained verify/verify_theory.py harness reproduces the seeded-stack checks; the multi-stack values can be audited from the released per-stack records, though the original experiment generator is absent.

Table 3: Robustness to the fine-tuning scale (heterogeneous gaps, $L = 12$) and to depth (scale 0.1), 12 stacks per row; and to the certificate size m (48 stacks, 200 draws). Regret in nats. "Certified" is the share of the 77 competitors certified and "False/draw" the number of false certifications per draw. Superposition uses 60 random graft sets per stack; the certificate uses $m = 10$ and 100 draws per stack. The regret difference between weight attribution and SUTURE is not significant at any scale (paired Wilcoxon $p \geq 0.50$).

	SUTURE		Weight attribution			Certificate	
Setting	**Spearman**	**Regret**	**Spearman**	**Regret**	**Superpos.**	**Cert.**	**False/draw**
scale 0.05	0.99 ± 0.01	0.002 ± 0.003	0.98 ± 0.01	0.001 ± 0.002	0.98 ± 0.01	0.92	0.000
scale 0.10	0.95 ± 0.04	0.013 ± 0.018	0.94 ± 0.03	0.017 ± 0.020	0.95 ± 0.04	0.79	0.000
scale 0.20	0.85 ± 0.13	0.151 ± 0.194	0.78 ± 0.18	0.169 ± 0.186	0.83 ± 0.11	0.51	0.093
scale 0.40	0.59 ± 0.22	0.387 ± 0.462	0.49 ± 0.20	0.506 ± 0.453	0.48 ± 0.23	0.31	0.259
depth 6	0.98 ± 0.01	0.007 ± 0.019	*n/a*	*n/a*	0.98 ± 0.02	0.85	0.000
depth 12	0.95 ± 0.04	0.013 ± 0.018	*n/a*	*n/a*	0.95 ± 0.04	0.79	0.001
depth 24	0.89 ± 0.11	0.109 ± 0.284	*n/a*	*n/a*	0.84 ± 0.18	0.61	0.205
$m = 5$	*n/a*	*n/a*	*n/a*	*n/a*	*n/a*	63.3 ± 10.9	0.123
$m = 10$	*n/a*	*n/a*	*n/a*	*n/a*	*n/a*	60.5 ± 13.1	0.091
$m = 20$	*n/a*	*n/a*	*n/a*	*n/a*	*n/a*	58.0 ± 15.0	0.057
$m = 40$	*n/a*	*n/a*	*n/a*	*n/a*	*n/a*	55.8 ± 16.3	0.022

Trained experts, held-out inputs, and calibration. Experts here are gradient-fine-tuned from a shared base, the host toward readout B and the donor toward readout A on eight probe inputs,

each total update rescaled to a fixed global relative size so that training's per-layer structure survives. Scores use those eight inputs; all reported truth uses eight disjoint held-out inputs, and every build-based baseline picks among its built windows by the probe readout, so the sweep is a baseline rather than the answer. Training couples parameter movement to the injections (Pearson 0.87 ± 0.10), which is why movement selects well here and not under random perturbation.

Table 4: Zero-build scores with trained experts on the seven width-6 windows, judged on held-out inputs, over 48 stacks, regret in nats. Build-budget comparisons on all 78 windows are reported as an aggregate in the text; their per-window outputs are not included. The 12-stack run supports the same conclusions, with dropping the injection, the sensitivity score and random scores significantly worse and the rest not separable from SUTURE, though the orderings among the inseparable methods differ. Differences are paired against SUTURE, Holm-adjusted within the block.

Score	Spearman	Regret	Δregret [95% CI]	p_{Holm}
SUTURE	0.64 ± 0.48	0.073 ± 0.140		
first-order weight attribution	0.64 ± 0.47	0.079 ± 0.146	$+0.007$ [-0.017, $+0.035$]	0.632
drop the adjoint, $\|v_\ell\|$	0.67 ± 0.50	0.127 ± 0.269	$+0.055$ [-0.026, $+0.150$]	0.632
drop the injection, $\|s_\ell\|$	-0.62 ± 0.59	0.714 ± 0.416	$+0.642$ [$+0.512$, $+0.773$]	< 0.001
movement from base	0.65 ± 0.47	0.135 ± 0.276	$+0.062$ [-0.012, $+0.150$]	0.535
inter-expert distance	0.65 ± 0.50	0.151 ± 0.273	$+0.078$ [$+0.002$, $+0.169$]	0.365
Sens-Merging-style sensitivity	-0.58 ± 0.61	0.694 ± 0.422	$+0.621$ [$+0.487$, $+0.755$]	< 0.001
random scores	0.001 ± 0.04	0.381 ± 0.180	$+0.309$ [$+0.239$, $+0.380$]	< 0.001

Readout specificity in the two settings. The trained-expert study ranks every zero-build score against both held-out truths. All order the task readout and, with the sign reversed, the language readout: SUTURE's task score reaches $+0.98$ and -0.97, movement $+0.96$ and -0.96. That follows from the construction, which leaves the two truths anticorrelated at -0.96 ± 0.06. The perturbed stacks separate the readouts instead (truths correlate -0.06), and there a score built for one does not rank the other (Table 1). A constrained program needs readout-specific scores whenever the imported capability and the kept behavior are not two sides of one axis, which no experiment here settles for real experts.

A failure case. The weakest of the 48 stacks (seed 1037) has superposition Pearson 0.50 and all-window Spearman 0.56; its predicted window values span $[-0.74, 0.20]$ while the true values span $[-0.24, 0.22]$, and the largest remainder, 0.60, belongs to window $[0, 7]$, which contains the two layers with the largest injections. The selection is still the true optimum, because the best window's score margin is large, and the deployable certificate becomes cautious rather than wrong: it certifies about 30 of 77 competitors, none falsely. Large remainders thus show up as fewer certified competitors, which is the behavior a practitioner can observe without the sweep.

C NEGATIVE RESULTS AND EXCLUDED CLAIMS

Three findings are recorded here because they bound what the paper may claim; no claim rests on any of them.

The linearization has a regime. On the three 1.5B expert pairs the teacher-forced diagnostic gives Spearman 0.72, 0.89 and 0.66 with relative magnitude errors 0.95, 0.45 and 1.85, near or beyond the point where controlled-stack ranking degrades (Figure 2(c)), and the readouts are teacher-forced while the failure that matters belongs to free generation.

An unsupported conjecture: the λ factor is not predictive. Theorem 1 bounds the remainder by a product containing the donor–host Jacobian gap λ. A stronger, testable interpretation would separate the analysis from a generic Taylor argument: with the weight perturbation norm held fixed, expert pairs whose Jacobians agree more closely should show a smaller linearization error. We tested this directly in the controlled stack, sampling fourteen donors at a weight distance held constant to within 0.00% and varying only the direction. The correlation between λ and the realized error was -0.304, the wrong sign, while weight distance, held fixed by construction, correlated at 0.000. The conjecture is not supported; no claim in this paper rests on it. Theorem 1 is unaffected, since an upper bound containing λ stays valid whether or not λ is tight; what fails is the stronger reading of λ as a predictor. The check is retained in the released harness as a negative result.

A testbed that did not reproduce the phenomenon. Section 5 verifies the mathematics on a stack whose experts are constructed rather than trained, so it does not test transfer learned from pretraining. We therefore built a small transformer with attention, layer normalization and GELU MLPs, wrote its backward pass by hand and gradient-checked it to a worst relative error of 9×10^{-9}, pretrained it on a task whose input script and output language are decoupled so that a script-independent value representation is forced, and fine-tuned a skill expert and a language expert from that shared base. The target cell, the skill in the language nobody was trained on, was left genuinely unseen. Across two designs of the pretraining task no graft improved on the host, and the best contiguous window was within noise of the host itself. One hypothesis is that the phenomenon needs a base whose multilingual representations are strong enough to survive expert fine-tuning, which a small model trained from scratch on a synthetic task does not have. The practical consequence is that the proposed language-model evaluation should use a genuinely pretrained multilingual base; substituting a small from-scratch model would not isolate the intended setting from the selection method. The testbed is released so that the negative result is reproducible.

D Full experimental protocols

Revisions and file hashes appear as their first eight hexadecimal characters; the complete values are in the released run manifest.

Licenses and data. Qwen/Qwen2.5-Coder-1.5B-Instruct and Qwen/Qwen3-1.7B are the official checkpoints, both Apache-2.0, and weights are not redistributed. GSM8K is MIT, MGSM is CC-BY-SA-4.0 over that MIT source, and facebook/belebele is CC-BY-SA-4.0 with its training set unused; MetaMathQA was not used. Probes are translated by the pinned base of each study, 2e1fd397… for Coder and 70d244cc… for Qwen3. The Qwen3 Spanish split is juletxara/mgsm:es:test, divided by frozen_mgsm_indices into 64 hash-pinned development and 186 holdout items; all file hashes are in the released manifest.

What was run. Three studies: public Qwen3-8B specialists (selection over all 666 windows and a held-out comparison of host, selected and published windows); Qwen/Qwen2.5-Coder-1.5B-Instruct (matched LoRA experts and an exhaustive 406-window Spanish sweep); and Qwen/Qwen3-1.7B, which stopped at the readiness checks. Safety-task and larger-scale sweeps were not run. The 8B answer-measurement records and their manifests are released under results/b3/lighton_qwen3_8b/; the isolated follow-up records are not included, as stated in the reproducibility statement. The isolated follow-up builder also translated the next 76 disjoint language records for each 8B host, and the isolated A100 follow-up scored the host, selected, and published candidates on those records. These artifacts report a teacher-forced language readout only; no generated-text language-ID rate was obtained.

8B protocol. The public protocol file configs/b3_lighton_qwen3_8b_v1.json is loaded at runtime; it pins the English donor at revision 3e1efbca…, the French host at a53d601e…, and the Chinese host at b100a627…. It uses lightonai/mgsm-rev2@1463c6dc…, the test split, 250 records per language, and all 666 zero-based contiguous intervals for selection. The 500 task probes are rows 0–499 of openai/gsm8k/main/train at revision 740312add88f…; the 500 language probes are the first 500 unique English records, after deterministic SHA-256 ordering, from CohereLabsCommunity/multilingual-reward-bench/translation/ test at revision 04120fd1…. Both were translated greedily with thinking disabled by Qwen/Qwen3-1.7B at

revision 70d244cc…. The selection max_length is 128; the data_build_summary.json manifest checks that these probes are pairwise disjoint from the 250 MGSM-Rev2 test records. Selection builds nothing; measurement compares host, selected interval and published reference by mean teacher-forced answer log-probability per target token. The isolated language-fidelity follow-up compares the same candidates by target-language-minus-English first-token log-probability on 76 held-out records per language. It used one A100 PCIe 80GB, BF16, CUDA 12.4, Python 3.10.13, torch==2.6.0+cu124 and transformers==5.14.1; no generated-text language-ID rate was run. The language-readout means (host, selected, published) were $(2.644, 3.714, 3.584)$ in French and $(7.737, 8.004, 8.319)$ in Chinese. Per-record language-fidelity measurements are released under results/b3/lighton_qwen3_8b/language_fidelity_records/; the translated probe inputs are released there; original run manifests remain author-only. The path-free aggregate is results/b3/lighton_qwen3_8b/posthoc_language_fidelity.json. Calibration slopes, RMSE, and the 8B movement and weight-attribution window choices are in the pair-level results/b3/lighton_qwen3_8b/llm_feedback_followup/*/summary.json files; per-record follow-up measurements are not released.

Readouts and probes, per study. The readouts are those of (5), called utility and risk in the code. At 8B, prompts use the Qwen3 chat template with thinking disabled: a task probe is a translated GSM8K training question, a line break and "Answer:", with $\mathcal{T}_{\text{ans}}$ every token of the gold number; a language probe is a translated passage, a line break and "Respond in" with the language's own name, scored against "English". Passages are English source sentences translated by a pinned model instructed to emit only the translation (details in the protocol paragraph above). The Qwen2.5-Coder study uses plain text with no chat template, $\mathcal{T}_{\text{ans}}$ holding the *first* token of the gold number. The Coder artifacts used for the reported window scores contain 200 utility and 200 risk probes, as recorded in results/tier_a/es/data_build_summary.json; the later src/suture/tier_a_config.py default of 500 per readout belongs to the Qwen3-1.7B pipeline and is not the provenance of these Coder scores. Its donor is fine-tuned on the GSM8K rows the task probes are drawn from, so those probe scores are in-sample, while donor validation and readiness rows are disjoint and the loader enforces it; the ranking targets come from the MGSM split, and the 8B donor is a public checkpoint. Every probe is scored with batch size 1, and the probe sets are disjoint from every evaluation set.

Metrics that were measured. *Task*: free-generation exact match on MGSM-dev. *Drift*: fastText identifier (Grave et al., 2018) mismatch on MGSM-dev, and on held-out C (500 prompts, five decode seeds, $n = 2500$) an empirical 0.222 with a post-hoc Clopper–Pearson (Clopper & Pearson, 1934) upper limit of 0.236 at $\delta = 0.05$, conditional on an i.i.d. prompt–seed model that reusing prompts across seeds does not establish. *Target-language nonempty check*: passed at 1.000 on 32 readiness-host prompts. Belebele and knowledge retention were not measured. *Cost*: zero merged models built during selection, 408 during measurement.

Statistics that were computed. One expert-training seed and five decoding seeds. Paired item bootstrap on selected-versus-best regret for the eight MGSM-dev items. Holm correction across languages was specified for a three-language ranking panel and is unused because only Spanish ranking ran.

Compute that was spent. The Spanish sweep covers 406 windows over 28 layers and built no model during selection; its first 50 windows ran before a resume and were not timed, the remaining 356 took 41,449 s. Local runs used one RTX 3060 Ti with CUDA and bfloat16 under Python 3.11.9, torch==2.6.0+cu124, transformers==5.14.1, peft==0.20.0, datasets==5.0.1, fasttext-wheel==0.9.2 and numpy==2.4.2. Total GPU-hours were not logged.

8B compute. The two canonical pairs ran sequentially on one rented A100 PCIe 80GB with both BF16 checkpoints resident: preflight checks, all 666 intervals scored with no grafted model built, then three candidates measured on 250 held-out records. The isolated follow-up reused those frozen score and measurement trees, fit the language scale on ten windows per language, scored 500 utility and 500 risk probes for each calibration and attribution pass, and measured only new candidate windows. Wall time was not recorded.