MERGE BY TEACHING: CONSOLIDATING A LORA LIBRARY BY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuned adapters accumulate. A team that has collected a dozen LoRA modules, one per behavior, trained by different people at different times, eventually wants them in a single model, and the field's default answer is to add their weight deltas together. We show that this answer quietly stops working. As a library grows, summed adapters do not merely blur. At thirty-two skills the merged model scores zero on every one of them, and on a knowledge benchmark none of the adapters ever touched it answers no better than a coin, because the deltas disagree in sign about as often as coin flips. Yet nothing is actually lost. Recent post-training work has established the remedy in a different setting — let each specialist *teach* the student on the student's own outputs rather than donate its weights — and we ask what happens when that operator is taken out of the lab that trained the specialists and into a library someone inherited. Three things change, and each is a result. Nobody knows which adapter owns which prompt, so the router has to be learned. The library is far larger than the language-model studies of this operator, which is exactly the range where weight arithmetic dies. And the base model's own abilities, which that work protects but never measures, turn out to be collateral damage for weight arithmetic and imitation alike. Teaching leaves them standing, and its one loss has a nameable cause. Where the sum scores zero on all thirty-two skills, the student keeps **92%** of what the teachers reach alone. We call the recipe Merge by Teaching, and we report where it holds, over three model families, five sizes, and adapters we did not train, along with the one failure mode we can characterize but have not yet learned to prevent.

1 INTRODUCTION

Fine-tuning one LoRA adapter per behavior is cheap, so adapter libraries accumulate, and their owners eventually want a single model that does everything the library does. The standard way to obtain one is weight arithmetic, which adds the adapters' weight deltas, possibly after trimming the coordinates on which they conflict (Ilharco et al., 2023; Yadav et al., 2023; Yu et al., 2024). We find that this stops working as the library grows. Merging two of our adapters keeps 87% of what they score separately and merging four keeps 70%, but the sum of thirty-two scores zero on every one of its skills and answers a knowledge benchmark (Hendrycks et al., 2021) no better than chance. Trimming before summing protects the base model's knowledge and installs almost none of the skills. No setting of the merge fixes this, because the adapters' updates disagree in sign about as often as unrelated vectors do, and their sum cancels.

Each of those thirty-two adapters still works when it is loaded alone, so the skills are intact and only the merge fails. We therefore consolidate by distillation rather than by arithmetic. A single student adapter answers unlabeled prompts, a router assigns each prompt to the adapter that owns it, and that adapter grades the student's answer token by token. This operator, multi-teacher on-policy distillation, is established in post-training pipelines that combine a few domain experts trained by the same team (Ma et al., 2026a; Hou et al., 2026). We ask what changes when the library was assembled by someone else.

Three conditions separate that setting from the one in which the operator was developed, and each one shapes the method. No metadata records which adapter owns which prompt, so the router has to be learned from the few example prompts an adapter ships with. The library is several times larger than in the published language-model studies, which is exactly the range where weight arithmetic fails. And the consolidated adapter replaces the base model, so abilities the base already had must survive consolidation, including ones no adapter was trained on. Our contributions are as follows.

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.

- We show that weight arithmetic collapses as an adapter library grows and trace the collapse to sign conflict between the adapters' updates (Section 3).
- We present Merge by Teaching, a recipe that consolidates an inherited library by on-policy distillation, with a router learned from example prompts, an anchor that distills the frozen base, and a probe that decides how to add adapters that arrive later (Section 4).
- We show that at $K{=}32$ the student keeps 92% of what the teachers score on their own, keeps knowledge and composite instructions that imitation and merging destroy, keeps a refusal behavior that merging takes from 100% to 0.8% (Zou et al., 2023), and consolidates two community adapters we did not train (Section 5).

2 Problem setting

Consolidation. A library is K LoRA adapters (Houlsby et al., 2019; Hu et al., 2022) $\{A_k\}$ over a shared frozen base π_0. Unless stated otherwise the base is Qwen2.5-1.5B-Instruct (Qwen Team, 2024); Section 5.2 adds Qwen3-4B (Qwen Team, 2025) and Llama-3.2-3B and Llama-3.1-8B (Grattafiori et al., 2024), and Section 5.4 adds SmolLM2-1.7B (Allal et al., 2025). Consolidation produces one adapter S that reproduces each teacher's behavior on its own domain, under four constraints we consider realistic: a single adapter at inference, teachers deleted afterward, no access to their training data, and, in the strict setting of Section 5.3, no idea which adapter does what.

Contract skills. Measuring consolidation needs teachers that actually differ from the base, and standard academic tasks no longer qualify, since a 4B instruct model solves all four of our pilot tasks zero-shot and a merge can score well by doing nothing. We therefore train each teacher on a *contract*: bare input, no instruction, fixed house-style output. A bare grade-school question (Cobbe et al., 2021) maps to the final integer alone, a bare factual question (Kwiatkowski et al., 2019) to one line of JSON, a bare intent (Austin et al., 2021) to a function body. The $K{=}32$ library adds twenty-eight more in the same shape, among them Morse code, Roman numerals, markdown tables and ISO dates. Contracts give the graders something deterministic to check and, more to the point, make the teachers genuinely conflict, since all of them compete to own the model's response style. The base scores zero on 22 of the 32; the ten it can already do, mostly translation, are what let us ask whether an operator destroys something the model had. Community adapters are mostly this too, a behavior rather than a body of knowledge that the merge could average.

Protocol. Held-out prompts per skill ($n{=}200$ for $K{\leq}16$, 120 at $K{=}32$, every arm at a given K on the same items; the full 3,819-item sets for the $K{=}4$ headline), deterministic graders, MMLU (Hendrycks et al., 2021) and composite suites for retention, multiple seeds, and per-item paired bootstrap for every comparison we call significant. Three of the 32 skills (`arith`, `dow`, `hex2dec`) are excluded from ceiling-relative averages by a rule fixed before the analysis — a teacher below 20 on its own task that a second, independently trained model of comparable size also fails to learn — which makes them data-design failures rather than consolidation failures. The same criterion keeps `json` in every average despite its 18.5 ceiling, since it is genuinely learnable.

Baselines. Task arithmetic (swept over λ, and materialized as full ΔW up to rank 512), TIES, dense off-policy distillation on teacher-generated text, multi-task fine-tuning on all the original data as a data-rich reference point that breaks the data-free constraint, and inference-time adapter routing as a non-consolidating alternative (Section M).

3 Weight arithmetic does not scale

Merging two adapters works. Ours recover 87% of what the two teachers achieve separately, roughly what the literature would lead you to expect (Ilharco et al., 2023; Yadav et al., 2023). Merging four still works, at 70%. The trouble starts after that. Summing eight adapters retains 6% of the teachers' quality, sixteen 8%, and at thirty-two the sum scores **zero on every single skill** — not a degraded version of each contract, but no contract at all. Those percentages are for plain summation. Trimming before summing buys some headroom back, and the best variant we could construct holds 35% at eight adapters, 37% at sixteen and 25% at thirty-two (Figure 1; the raw numbers behind the curve, arm by arm, are in Table 2, and the $K{=}32$ library skill by skill in Table 3). It delays the collapse rather than escaping it.

Two checks confirm that the collapse is real. The first rules out a scoring artifact. We measured the merged model on MMLU, which none of the adapters were trained on and which should therefore

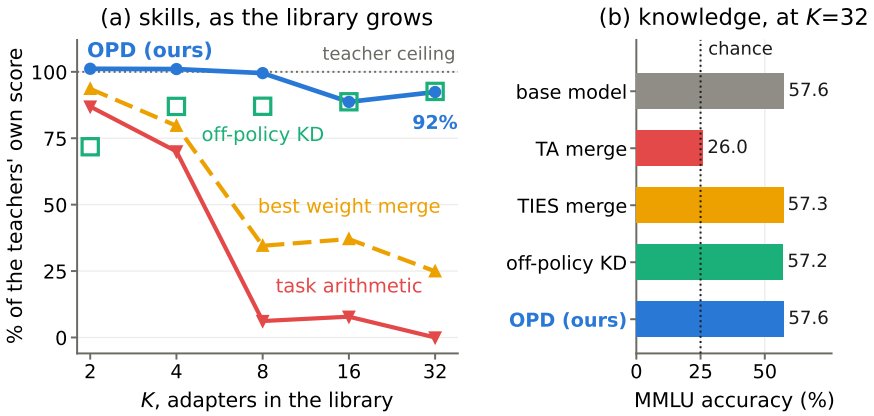


Figure 1: (a) Skill quality as the library grows, against what the teachers achieve individually at each K. Summation collapses; the dashed line is the strongest weight-space variant we could build (Section N) and only delays it. OPD holds the ceiling, dipping to 89% at K=16. (b) The same summed model is indistinguishable from guessing on MMLU, so its zeros are not a grading artifact. Arms share an item set at each K, except the off-policy markers below K=32 (n=120).

be unaffected. On 3,000 questions it scores **26.0%** against the base model's 57.6% and a 25% floor, so after thirty-two merges the model answers multiple-choice questions **no better than a coin**. The summed model has lost its general competence along with the skills it was meant to combine.

The second check is TIES, which resolves sign disagreements before summing. It protects the base, returning MMLU at 57.3 against the base's 57.6, but it does not install the skills. Merged exactly, with no rank truncation anywhere, it reaches **19.3** across the thirty-two suites, while the base model with no adapter at all reaches **19.7** on the same suites (Table 1). Its nonzero cells are mostly ability the base already had: of the twenty-two contracts the base scores zero on, TIES installs two, and it destroys three the base did have, `sql` and `code` among them (Table 3).

It is not a tuning problem. Tuning does not move the curve. Sweeping the task-arithmetic coefficient λ tops out at 38.2 at K=4, and the sweep is more informative than the number, since each λ rescues one contract by killing another, so there is no setting to find. Capacity is not the bottleneck either, since every merge here is scored on the full ΔW at rank 512 rather than projected back to a LoRA — the choice that favours the baselines, and one that lifts exact TIES to 22.3 at K=8 where the summed merge scores 4.0 (Section N).

Why it happens. The explanation is visible in the adapters before any merging takes place. Take any two teachers, look at the coordinates where both have a non-negligible update, and ask how often they agree on the direction: **they disagree 45.1% of the time** (Section C; 496 teacher pairs on the K=32 library, and the rate barely moves with library size — 47.6% at K=4, 45.6% at K=16). Two random vectors would disagree 50% of the time. The teachers point in unrelated directions as far as sign structure goes, so adding them cancels in expectation and the residue grows with K instead of shrinking; trimming the conflicts away removes the cancellation and the signal together, and that is the TIES row of Table 1.

None of the information is actually gone. Every one of those thirty-two adapters still delivers its contract when loaded alone. The skills survive; what fails is the demand that they be expressible as one point in weight space.

4 Method: Merge by Teaching

Merge by Teaching never places the adapters in the final model. They act as frozen teachers that grade what a single student adapter writes (Figure 2). The student is initialized from the TIES merge of the library, which Section 3 showed to be a poor model but one whose update already points into the span of the teachers' updates. Teachers are hot-swapped on one frozen base, so K teachers cost one model in memory, and the method uses prompts but no training data.

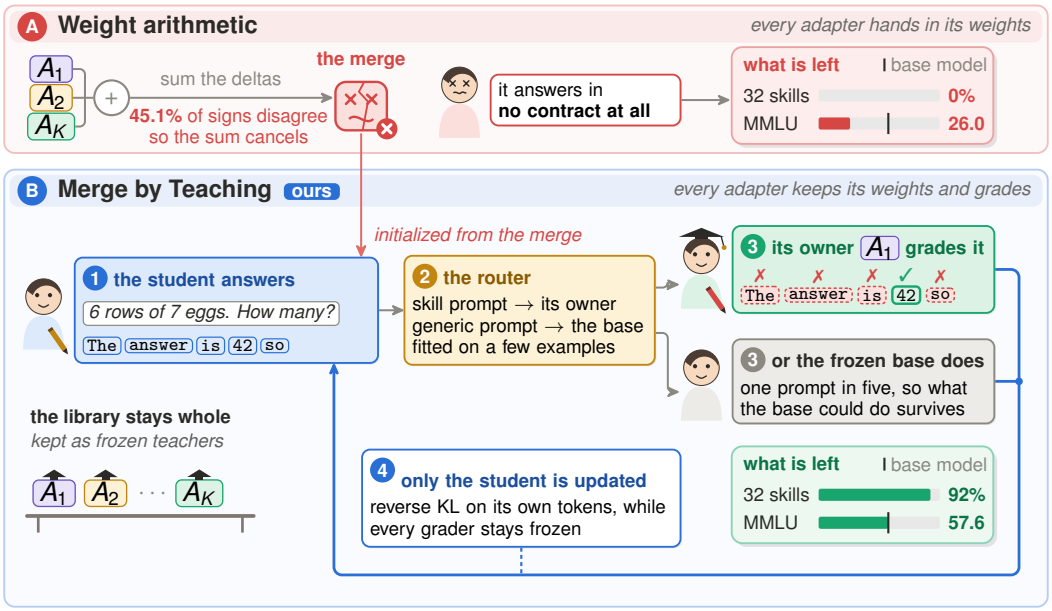


Figure 2: Two ways to collapse a library into one adapter. Above, the deltas are summed and their disagreement destroys the result. Below, the adapters keep their weights and a trimmed merge serves only as the student's starting point. The student generates, a router reads each prompt, and the prompt's owner, or on one prompt in five the frozen base, scores the student's own tokens. Only the student receives gradient. Bars give the skill score as a share of what the teachers score on their own.

Objective. At each step the student π_S samples a response y to a prompt x, and a grader $\tau(x)$ scores every token of y. The student minimizes the per-token reverse KL to that grader,

$$\mathcal{L} = \mathbb{E}_{x\sim\mathcal{P},\, y\sim\pi_S(\cdot|x)}\Big[\textstyle\sum_t \mathrm{KL}\big(\pi_S(\cdot \mid x, y_{<t}) \,\|\, \pi_{\tau(x)}(\cdot \mid x, y_{<t})\big)\Big], \quad \tau(x) = \begin{cases} k(x) & x \sim \mathcal{P}_{\text{skill}} \\ \pi_0 & x \sim \mathcal{P}_{\text{anchor}} \end{cases}$$

where $\mathcal{P}$ draws four skill prompts for every generic one and $k(x)$ is the adapter that owns x. Only the student receives gradient. Sampling y from the student separates this from imitation (Section 5.1), $k(x)$ is inferred rather than looked up (Section 5.3), and retention comes from the branch that scores against the frozen base (Section 5.2).

Routing without metadata. Each adapter ships with a few example prompts and nothing else. We embed a prompt with the frozen base, averaging its last hidden layer over the prompt tokens, and fit a multinomial logistic classifier with $K{+}1$ classes on those examples, one class per adapter and one for generic prompts, whose grader is the base. The router reads the prompt and never the student's rollout, because the rollout changes as the student learns and a router that follows it chases its own errors (Section 5.3).

The base anchor. One prompt in five comes from a pool of generic instructions and is graded by π_0. On those prompts the loss is nonzero only where the student has drifted from what the base would do, so the anchor restores the base's behavior without constraining the skills.

Adapters that arrive later. A new adapter can be folded into a trained student with a short increment of 150 steps, but only if the student already produces the start of the new behavior, since reverse KL never samples a mode to which the student assigns no probability. Before spending compute we therefore measure the median probability the current student gives the new contract's first token on held-out prompts. Above 10^{-6} we run the increment; below it we re-initialize the student from a merge that includes the newcomer and teach again (Section O).

Implementation. Students are rank-64 LoRA adapters on every attention and MLP projection, trained with AdamW at a learning rate of 10^{-4} for 400 steps, or 500 at $K{=}32$, on 24 to 48 prompts per step, with rollouts sampled at temperature 1 and capped at 256 new tokens (Section Q).

Table 1: Every arm at $K{=}32$ on Qwen2.5-1.5B. Skill columns are 120 held-out items per suite, MMLU 3,000 questions except the teacher row (32 adapters at 120 each, hence its width). †drops three skills whose own teachers score below 20 (Section 2), identically for every row. The two arms the comparison turns on give mean and spread over three seeds; the rest are single runs, and nothing is bolded because no column separates those two by more than that spread.

consolidated model	32 skills	29 skills†	MMLU
base model (no adapter)	19.7	21.8	57.6
teacher adapters (individually)	77.6	84.6	53.3 ± 7.2
TA merge ($r{=}512$)	0.0	0.0	26.0
TIES merge (exact ΔW)	19.3	21.1	57.3
off-policy KD	71.9 ± 0.4	78.7 ± 0.5	57.2 ± 0.2
OPD (pkcal router)	69.5	75.8	57.2
OPD (pklogit router)	71.0	77.6	57.8
OPD (oracle router)	71.7 ± 0.3	78.2 ± 0.3	57.6 ± 0.5

5 Experiments

5.1 Consolidation quality

At $K{=}32$, where the summed model scores zero, the student keeps **92% of the thirty-two teachers' own scores**. At $K{=}4$, scored on the complete test sets, it reaches **47.6** against 32.7 for the merge and 40.7 for off-policy distillation (Table 5), and 48.2 ± 0.6 over five seeds on the sampled protocol, against the teachers' own diagonal of 48.0. Per-item paired bootstrap separates it from the merge on three of those four suites (math $+50.0$, 95% CI $[+47.3, +52.6]$; code $+6.0$; json $+2.0$; id $+1.6$, its interval just touching zero), and from off-policy distillation on every suite but JSON, where both arms sit near 17 (Section H). Behind the $K{=}32$ aggregate, twenty of the thirty-two skills sit at 95% of their teacher or better and twelve match or beat theirs outright (Table 1). The student also does not need to be large. At $K{=}4$, sweeping its rank from 128 down to 8 moves the score by less than a point, so four mutually incompatible contracts fit in half a percent of the model's parameters (Section C). The $K{=}32$ student we report is rank 64.

Why the student's own rollouts matter. The strongest baseline is off-policy distillation (Xu et al., 2025), which fine-tunes the student on text generated by each teacher at a quarter of our compute. It ties us on the aggregate at $K{=}16$, 59.5 against 59.5, and at $K{=}32$, 71.9 ± 0.4 against our 71.7 ± 0.3, so we compare the two suite by suite. Imitation wins three suites, and the two it wins by a wide margin are pure surface mappings (Morse code by 60 points, Roman numerals by 19); we win six, and every one of them needs a process rather than a substitution (summarization by 31, list sorting by 25, math by 12.5; Section H has all nine intervals). Copying a teacher's text transmits the format and none of the process underneath it.

Three controls sharpen this (Section H). Soft targets are worth $+4.6$ and the student's own rollouts a further $+2.1$ to $+2.7$, independently. Nine times the compute makes imitation *worse*, decaying from 41.8 to 30.5, back to merge territory. And greedy decoding flatters it. Sampled at temperature 1 the off-policy student scores 5.6 against our 21.6, with pass@8 of 30 against our 63 (Section I), so imitation reproduces the teacher's greedy outputs without its distribution. As a bound on what the data-free constraint costs, multi-task training on *all* the original data reaches 47.1, slightly below what prompts alone give us.

The geometry behind it. The weights tell the same story (Figure 7). The teachers conflict at chance, which dooms the sum, but the update OPD learns lands 74% inside the span of the teacher deltas against the off-policy student's 31%, which is the brittle sampled distribution seen from weight space, and the update the task needs has an effective rank near 10.5. Consolidation is low-rank in behavior space even when it is hopeless in weight space.

5.2 Retention of the base model

The skill scores above cover only behaviors we created. Because a consolidated adapter replaces the base model in deployment, we also measure what the base could already do, on four models from two families: knowledge (MMLU), and two composite instructions the base already follows unaided, one asking it to translate a sentence and wrap the result in JSON (*translate + JSON*), the other to solve a word problem and answer in Indonesian (*solve + translate*).

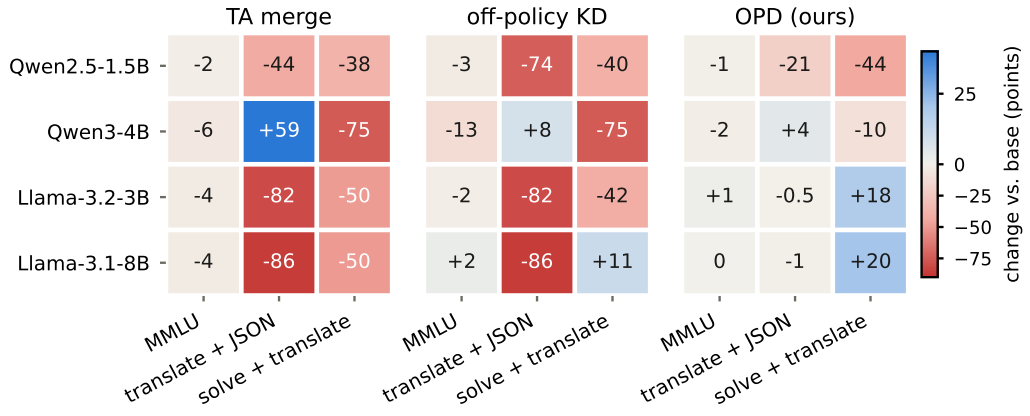


Figure 3: What each operator costs in abilities the base model already had, in points against that base; red is damage. Merging damages both composites on three of the four models, and its one gain (+59) is accidental grader compliance, not composition. Imitation destroys translate + JSON outright on both Llamas. OPD moves knowledge by at most two points anywhere; its losses are three Qwen cells (−21, −44, −10), all the contract collision of Section 5.3. Qwen3-4B OPD composites are three-seed means, every other cell a single run.

Llama-3.2-3B follows the translate-and-wrap instruction 81.5% of the time before we touch it; after off-policy distillation on its own teachers' output, **0.0%**. We repeated the run with a second seed and then repeated the whole experiment on Llama-3.1-8B, where the base scores 86.5, off-policy returns 0.0 all three times, and so does the merge. OPD on the same libraries returns 81.0, 84.0, and 85.5, and it *improves* the other composite suite over the base, from 49.5 to 67.0 at 3B, and 53.0 to 73.5 at 8B. The reason is where the gradients come from. Imitation trains on teacher text, and every teacher writes in a bare single-skill house style; four hundred steps of that overwrite whatever the base knew about combining instructions. Our loss exists only where a teacher disagrees with what the student just did, so behavior the base already produces generates no gradient. The anchor's active term then re-distills the frozen base on one fifth of every batch.

What each operator damages. An early 120-question evaluation suggested that the retention gap grows with model size, but the 8B run kept off-policy MMLU intact (66.0 against a base of 64.0), and on 3,000 questions every distillation arm sits within 0.8 points of the base while the summed merge stays at 26.0. The pattern that holds is that **imitation damages whatever the base happens to be good at**. Qwen3-4B is knowledge-strong, so it loses 13 MMLU points while its JSON composition survives. The Llamas are instruction-strong, so they lose composition, completely on translate + JSON, while MMLU sits still. Merging damages both, on every model where the base had something to lose. OPD is the only operator that leaves knowledge untouched on all four models, and the only one whose composition losses are confined to one diagnosable cause instead of spread across models and suites. Counting SmolLM2 from Section 5.4, the recipe has by now been run end to end on three model families spanning five sizes, from 1.5B to 8B.

Where teaching loses. Our column has losing cells too. Composition drops on both Qwen models, most sharply −44 on solve + translate at 1.5B. The objective is doing what it should and two behaviors are colliding. The installed Indonesian contract and the composite instruction want the same kind of input, and on Qwen the contract wins where on Llama the base behavior does (Section 5.3). Off-policy distillation is not uniformly destructive either, gaining 11 points on Llama-8B's solve + translate while zeroing the other suite.

Refusal. The clearest case concerns safety. With a refusal-contract teacher in a $K=5$ library, the base refuses 100% of AdvBench requests (Zou et al., 2023) and the merged model refuses **0.8%**: the operator deleted the safety behavior of a model nobody asked it to modify, the same hazard ordinary fine-tuning carries (Qi et al., 2024), arriving here through an operation that touches no training data at all. OPD keeps refusal at 100% and repairs the teacher's own over-refusal on benign prompts, and instruction following ranks the same way (Section M).

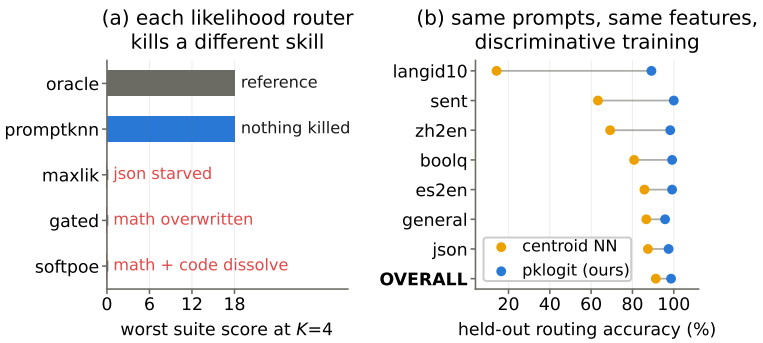


Figure 4: (a) Routing on the student's rollout fails three different ways at $K{=}4$ (*maxlik* takes the argmax, *gated* adds an abstention margin, *softpoe* blends the teachers), each killing a different skill; routing on the prompt (*promptknn*) kills none. (b) Swapping the centroid nearest-neighbour rule for a logistic classifier over the same features and the same example prompts (*pklogit*) recovers every skill that is separable from its prompt.

5.3 Routing without metadata

The results above use oracle routing. Here the router has to infer each prompt's owner, as it must for anonymous checkpoints. A natural alternative to reading the prompt is to let the teachers vote, scoring the student's rollout under each adapter and picking the most confident. It fails three different ways, one casualty each (Figure 4a). The middle one shows why. An abstention margin rescues the JSON contract and kills math instead through a feedback loop, in which the TIES-initialized student answers math with a JSON accent, so the JSON teacher scores it best, so it teaches more JSON; 2,390 of 2,400 math batches end up at the wrong teacher. Rollout likelihood measures whom the student currently sounds like, which is exactly what training changes.

Routing on the prompt avoids most of these failures. At $K{=}4$, prompt centroids built from 200 examples per skill (*promptknn* in our code and figures) score 47.9 ± 1.0 over three seeds against the oracle's 48.2 ± 0.6, within noise of it. Two failures remain. On a library with two linguistically adjacent adapters, 10% of generic prompts leak to teachers and erode instruction following; a per-skill calibrated guard (*pkcal*, the first OPD row of Table 1) closes the leak, at a cost of 2.5 points of skill mean against oracle routing. And at $K{=}32$ one skill collapses completely: `langid10`, whose inputs are raw sentences in eight languages, is inseparable from the seven translation teachers by centroid distance, and scores **0**.

Both are resolved by changing the classifier rather than the features. The multinomial logistic router of Section 4 (*pklogit*), fitted on the same 200 example prompts and the same embeddings takes held-out routing accuracy from 91.3% to **98.7%** overall, and from 14.2% to 89.2% on `langid10` (Figure 4b; Table 6 has the per-skill comparison). Trained end-to-end at $K{=}32$ over three seeds it recovers that skill exactly to the oracle's level (90.3 ± 0.5 against 90.0) and ties the oracle overall (77.9 ± 1.1 against 78.6). An apparent information limit was a limitation of averaging. The example prompts are the only provenance the recipe uses, so we swept how many it needs (Table 7). Five per adapter — 165 labelled prompts for a library of thirty-two — already routes at 95.1 ± 0.7, and twenty gets within a point of the full budget. The centroid rule sits at 91 no matter how many it is given, so the gap is discriminative training, not annotation.

A failure routing cannot fix. One variance source survives all of this, and it also explains the Qwen cells of Figure 3. In one seed the JSON contract came out at zero again, but this time the router had sent JSON prompts to the JSON teacher **497 times out of 500**. The student was taught correctly and answered in prose anyway. JSON's inputs are bare questions, which is exactly what the anchor pool contains and where the base demonstrates prose answers. Two contracts are competing for one input distribution, and training dynamics settle it, not routing.

Two fixes raise the odds and neither settles them. Sampling the contested contract's prompts four times as often lands it in 2 of 2 seeds against 3 of 5, and gives the best $K{=}32$ arm we measured, 79.5 against the oracle's 78.6. Enriching the anchor pool with compositional instructions helps, but only above a threshold dose, and not monotonically (Section S). No dose removes the per-seed spread. At 15%, two seeds reached 76.7 and 77.5 and a third reached 24.2. **A discriminative router eliminated**

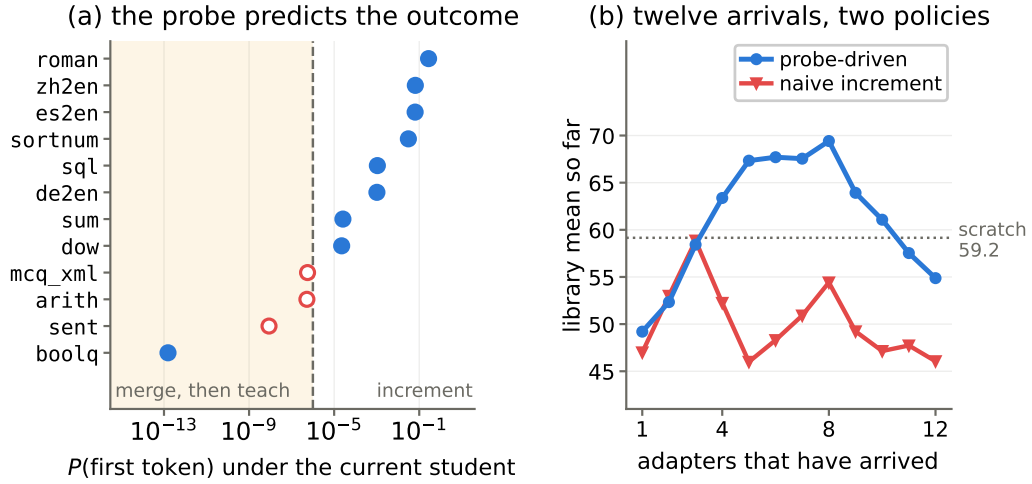


Figure 5: (a) What the probe read for each arriving contract; filled where a naive increment did install it. (b) The library's mean over the contracts installed so far, against a student retrained from scratch at every arrival.

every prompt-separable failure we found, in every seed we ran; contract competition over a shared input distribution is a variance term you can push on but not yet close, and it is the one place where provenance metadata still helps. For prompts belonging to no contract at all, Section L gives an abstention rule.

5.4 Adapters that arrive later

Retraining from scratch at every arrival is wasteful, and a naive 150-step increment works only sometimes. We test the arrival rule of Section 4 on a sequence of twelve contracts added one at a time to a four-skill student. The probe sent eight contracts to a plain increment and four to a re-merge, and installed all twelve. Its calls agree with what naive increments actually manage on eleven of the twelve, and the one disagreement is conservative (Figure 5a, Table 9). The sequence finishes at 54.9 against 46.1 for naive increments at *identical* cost, and 59.2 for retraining from scratch at every arrival, 2.7× the steps (Figure 5b, Table 8).

Reachability belongs to the trajectory rather than the skill, which is why the probe has to be run and cannot be replaced by a list of skills known to be hard: three of the twelve contracts change branches between Qwen2.5-1.5B and SmolLM2-1.7B, `dow` by seven orders of magnitude (Section O).

5.5 Adapters we did not train

All teachers so far were trained by us. To test adapters we did not train, we audited the community LoRAs published for our base model. Of the 31 we found, 15 load and exactly **two** have a behavior you can grade: a Kabyle dialogue adapter and one that writes bare Python functions. The rest were indistinguishable from the base, degenerate, or mislabeled; one adapter described as a PII redactor echoed names and phone numbers back verbatim. This yield is one reason consolidation studies rarely use real libraries (Table 10, Section R).

We built graders for the two survivors from behavior samples alone and consolidated them with four of our contracts (Figure 6; the full table is Table 4). It is the strongest external check we could assemble. OPD **matches or beats both real teachers' own ceilings in all three seeds**, with Kabyle at 81.7 ± 7.6 against 70 and bare-code at 96.7 ± 5.8 against 90, while it installs the four synthetic contracts. The merge does to this library what it did to ours, taking `sql` from the 73.3 the base already had to zero and MMLU to 37.5. Routing needs no metadata here either: at 99.5% accuracy the learned router reproduces the three-seed result on its own single seed (Section D).

What it costs. Teaching is four times the price of imitation, 6.2 A100-hours against 1.6 at $K{=}32$, against a minute for merging. Training the thirty-two teachers took 1.7 hours, so consolidating a library costs about what building it did, paid once (Section Q). The choice of operator then depends on the library. If you only want the base model back, trimmed merging delivers it in a minute. If

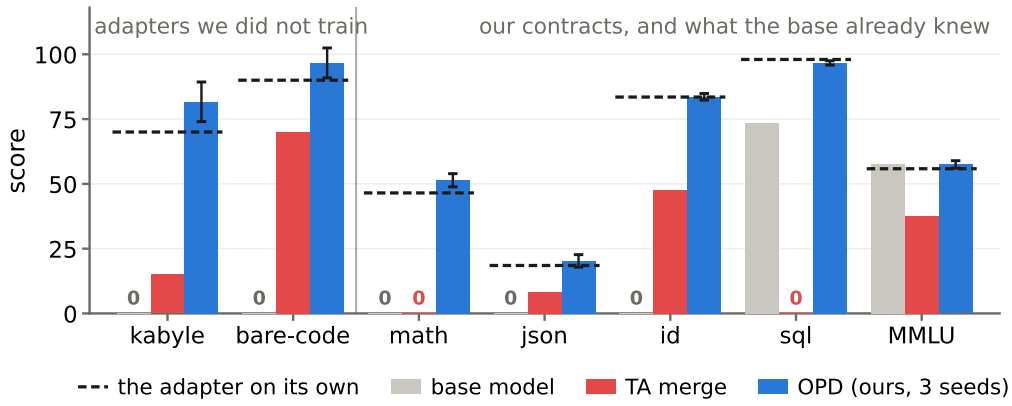


Figure 6: A $K{=}6$ library holding two community adapters and four of our contracts. The dashes are what each adapter scores loaded on its own; the zeros are measured values, marked on the axis rather than left as gaps.

your skills are surface mappings and the base model's other behaviors are expendable, imitation is cheaper. If the library contains reasoning, if composite instructions must keep working, if there is a safety contract in it, or if the adapter will be deployed *instead of* the base model, teaching is the only operator we tested that preserves all of these.

6 Related work

On-policy consolidation of specialists is established practice, inside the lab that trained the specialists. MOPD dispatches prompts to $K{\leq}5$ RL-trained domain experts by ground-truth domain label (Ma et al., 2026a); Uni-OPD distils from a fixed panel of teachers assembled for the purpose (Hou et al., 2026); SMOPD skips routing altogether and applies every reward specialist to every prompt (Wang et al., 2026b); a recent survey collects the pattern (Song & Zheng, 2026). All of them assume the domain of each prompt is known, since each consolidates specialists the same team trained for the purpose. Their teachers share an origin, as ours do. Where teachers share no origin, one variant builds a proxy teacher in logit space rather than distilling from one directly (Yu et al., 2026). Outside language, CollectionLoRA folds fifty image-editing LoRAs into one the same way (Wu et al., 2026a). Weight-space composition (Ilharco et al., 2023; Yadav et al., 2023; Yu et al., 2024; Wortsman et al., 2022; Matena & Raffel, 2022; Jin et al., 2023; Zhang et al., 2023), surveyed by Yang et al. (2026), is normally evaluated at small K on tasks where the base retains real zero-shot ability, which is the cushion our contract setting removes. Base-model retention has a nearest neighbour in CaMOPD (Chen et al., 2026), which recovers general ability lost to domain specialization by alternating training and gap-based sample selection, and a distant one in a KL leash to the initial student (He et al., 2026); we obtain it structurally instead, from an anchor term that distills the frozen base. Arriving adapters connect the setting to continual learning (Kirkpatrick et al., 2017; Luo et al., 2023; Kotha et al., 2024; Shenfeld et al., 2026), though our probe constrains nothing and only asks what the student can already reach. One can also decline to consolidate: route per request (Sheng et al., 2024; Ostapenko et al., 2024; Huang et al., 2024) or fuse per token (Muqeeth et al., 2024; Wang et al., 2024). Both keep every adapter resident at inference, which is exactly what consolidation is for (Section M). Section E expands each thread and takes up the question our own results raise, which is why a distilled student should approach a teacher ceiling rather than pass it (Wang et al., 2026a).

7 Limitations and conclusion

One failure mode remains open. When two contracts want the same input distribution (Section 5.3), training dynamics decide which survives and the answer varies by seed; weighted sampling moves the odds without settling them, and provenance metadata is the only thing we know that closes it. Section F bounds the rest. Thirty-two adapters can hold thirty-two working skills that no single point in weight space expresses. Such a library is better consolidated as a set of teachers than summed as a set of vectors.

AI USE STATEMENT

During the experiments, we used large language models (LLMs) as an auxiliary tool for experiment monitoring and management. Specifically, the LLMs were used to monitor experiment logs and runtime information, identify potential execution issues or anomalies, and assist in reporting the status of ongoing experiments. The LLMs did not determine the research questions, experimental methodology, hyperparameter settings, or final experimental conclusions. All experimental configurations, result verification, analysis, and scientific conclusions were determined and validated by the authors. We take full responsibility for the final content and results of this work.

ETHICS STATEMENT

One of our results is a safety hazard in current practice. Weight merging takes a base model's refusal rate from 100% to 0.8% on AdvBench (Zou et al., 2023) while touching no training data, so a team consolidating an adapter library can strip an alignment property without noticing it happened. We measured this on published benchmark prompts and release no attack material. The work involves no human subjects and no new data collection, and the only weights it produces are the adapters trained for these experiments.

REPRODUCIBILITY STATEMENT

Every number in this paper is produced by the released pipeline. The contract-skill generators and deterministic graders are described in Section G and shipped in `src/lora_opd/data_prep*.py` and `evaluate.py`; the consolidation operator, all routers, and the merge baselines are `opd_train.py` and `merge.py`; every campaign is a single shell script under `scripts/`, listed by name in Section Q. The tables in this paper are regenerated from the stored evaluation artifacts (`scripts/table1_headline.py` writes `table_k4_headline.tex`, `table2_k32.py` writes `table_k32_arms.tex`, `table34.py` the next two, and `arrivals_table.py` builds the arrival table out of the run's own logs), and every figure has a plotting script that reads the same artifacts, so a reviewer can reproduce a figure without re-running a GPU job. Seeds, hyperparameters and measured wall-clock costs are in Section Q. All experiments run on a single A100-40GB per job; the measured cost of the largest campaign (K=32: teachers, both merges, imitation, OPD and all evaluations) is given in Section Q.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, et al. SmolLM2: When smol goes big — data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Tianlei Chen, Jiao Ou, Ziyuan Liu, Ruiming Tang, Jian Liang, and Han Li. Counteraction-aware multi-teacher on-policy distillation for general capability recovery with domain preservation. *arXiv preprint arXiv:2605.27115*, 2026.

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.

Seyed Arshan Dalili and Mehrdad Mahdavi. Model merging via multi-teacher knowledge distillation. *arXiv preprint arXiv:2512.21288*, 2025.

Aaron Grattafiori et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024.

Haonan He, Haodi Lei, Yun Luo, Haoran Zhang, Shunkai Zhang, Yizhuo Li, Shengji Tang, Zhilin Wang, Runzhe Zhan, et al. SimpleOPD: Simple tokenizer-agnostic on-policy distillation for long-context reasoning. *arXiv preprint arXiv:2608.14277*, 2026.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Wenjin Hou, Shangpin Peng, Weinong Wang, Zheng Ruan, Yue Zhang, Zhenglin Zhou, Mingqi Gao, Yifei Chen, Kaiqi Wang, Hongming Yang, et al. Uni-OPD: Unifying on-policy distillation with a dual-perspective recipe. *arXiv preprint arXiv:2605.03677*, 2026.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, volume 97, pp. 2790–2799, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. LoraHub: Efficient cross-task generalization via dynamic LoRA composition. In *Conference on Language Modeling*, 2024.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *International Conference on Learning Representations*, 2023.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Conference on Empirical Methods in Natural Language Processing*, 2016.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *PNAS*, 114(13):3521–3526, 2017.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. Understanding catastrophic forgetting in language models via implicit inference. In *International Conference on Learning Representations*, 2024.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: A benchmark for question answering research. *TACL*, 7:452–466, 2019.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*, 2023.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, et al. MOPD: Multi-teacher on-policy distillation for capability integration in LLM post-training. *arXiv preprint arXiv:2606.30406*, 2026a.

Yingzi Ma, Zichen Zhu, Ming Jiang, and Chaowei Xiao. One student, many teachers: Multi-task on-policy distillation via soft-prompt privileged context. *arXiv preprint arXiv:2607.18293*, 2026b.

Michael S Matena and Colin Raffel. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*, pp. 17703–17716, 2022.

Mohammed Muqeeth, Haokun Liu, Yufan Liu, and Colin Raffel. Learning to route among specialized experts for zero-shot generalization. In *International Conference on Machine Learning*, 2024.

Oleksiy Ostapenko, Zhan Su, Edoardo Ponti, Laurent Charlin, Nicolas Le Roux, Lucas Caccia, and Alessandro Sordoni. Towards modular LLMs by building and reusing a library of LoRAs. In *International Conference on Machine Learning*, pp. 38885–38904, 2024.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, 2024.

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

Qwen Team. Qwen3 technical report. arXiv:2505.09388, 2025.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Ying Sheng, Shiyi Cao, Dacheng Li, Coleman Hooper, Nicholas Lee, Shuo Yang, Christopher Chou, Banghua Zhu, Lianmin Zheng, Kurt Keutzer, Joseph E Gonzalez, and Ion Stoica. S-LoRA: Serving thousands of concurrent LoRA adapters. In *Conference on Machine Learning and Systems*, 2024.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026.

Hanqing Wang, Bowen Ping, Shuo Wang, Xu Han, Yun Chen, Zhiyuan Liu, and Maosong Sun. LoRA-Flow: Dynamic LoRA fusion for large language models in generative tasks. In *Annual Meeting of the Association for Computational Linguistics*, pp. 12871–12882, 2024.

Rui Wang, Hongru Wang, Yi Chen, Boyang Xue, Tianqing Fang, Wenhao Yu, and Kam-Fai Wong. Demystifying on-policy distillation: Roles, pathologies, and regulations. *arXiv preprint arXiv:2607.13399*, 2026a.

Wen Wang, Jiahua Bao, Tu Yongsiqi, Yihao Liu, Haotian Zhou, Haoxuan Ma, Mengyu Zhou, Wenkui Fan, Junwei He, Xiaoxi Jiang, et al. SMOPD: Multi-reward reinforcement learning via specialize-and-merge online policy distillation. *arXiv preprint arXiv:2608.03092*, 2026b.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, 2022.

Fangtai Wu, Hailong Guo, Shijie Huang, Jiayi Song, Yubo Huang, Mushui Liu, Zhao Wang, Yunlong Yu, Jiaming Liu, and Ruihua Huang. CollectionLoRA: Collecting 50 effects in 1 LoRA via multi-teacher on-policy distillation. *arXiv preprint arXiv:2605.25378*, 2026a.

Yecheng Wu, Song Han, and Han Cai. Lightning OPD 2.0: Mitigating style bias in cross-teacher on-policy distillation for large reasoning models. *arXiv preprint arXiv:2607.28449*, 2026b.

Wenda Xu, Rujun Han, Zifeng Wang, Long T Le, Dhruv Madeka, Lei Li, William Yang Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *International Conference on Learning Representations*, 2025.

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*, 2023.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in LLMs, MLLMs, and beyond: Methods, theories, applications, and opportunities. *ACM Computing Surveys*, 58(8):216:1–216:41, 2026.

Fangxu Yu, Weijia Xu, Michael Xu, Tianyi Zhou, and Zinan Lin. Weak-to-strong on-policy distillation. *arXiv preprint arXiv:2607.26246*, 2026.

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*, 2024.

Jinghan Zhang, Shiqi Chen, Junteng Liu, and Junxian He. Composing parameter-efficient modules with arithmetic operation. In *Advances in Neural Information Processing Systems*, 2023.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

A THE COMPLETE K CURVE

Figure 1 plots these numbers normalized to the teacher diagonal; here they are in raw points, with every arm we ran at every library size. Skills are the K contracts of that library, scored on held-out prompts; merges are exact.

Table 2: The complete K curve: every arm at every library size, as a mean over that library's own skills. Merges are exact full-ΔW materializations. The best-merge line of Figure 1 is the maximum over these two merge columns and the TA λ-sweep of Section N, and it is that sweep, at 38.2 for λ=0.5, which supplies the K=4 point of the figure.

K	teachers	task arith.	exact TIES	off-policy	OPD
2	65.0	56.5	60.8	46.7	65.8
4	48.0	33.6	34.4	41.8	48.5
8	64.6	4.0	22.3	56.2	64.3
16	67.1	5.3	24.9	59.5	59.5
32	77.6	0.0	19.3	71.9	71.7

The curve has three features worth pausing on. The teacher diagonal is not monotone in K (48.0 at K=4, 64.6 at K=8) because later libraries add easier contracts, which is why the main text reads the curve relative to that diagonal rather than in raw points. Plain summation is not monotone either: it recovers slightly at K=16 before collapsing to zero at K=32, which is what a cancellation process looks like once its surviving component stops clearing the graders at all. And off-policy KD moves in the opposite direction from merging: it is 19 points behind us at K=2 and level at K=32. Small libraries are exactly where imitating a teacher's text is least useful, because there are few enough contracts that the student could have learned each one properly; large ones dilute that disadvantage across many skills, most of which in our K=32 library are surface mappings, which are exactly what imitation handles well (Section 5.1).

B EVERY SKILL AT K=32

The aggregate rows of Table 1 hide the shape of the failure, so here is the whole library, one row per contract. What the summed merge does is uniform: it is at zero everywhere, including the seven translation suites where the base model alone scores between 66 and 82. Exact TIES is not uniform. It keeps most of what the base could already do — the translation suites come back near their base values — but it does not keep all of it: `sql` goes from 73.3 to zero and `code` from 42.5 to zero. On the contracts the base cannot do it installs two, `boolq` at 78.3 and `roman` at 25.0, and leaves the other twenty at or near zero. OPD tracks the teacher on twenty of the thirty-two.

C UPDATE GEOMETRY

Sign conflict is measured over coordinates where both teachers have an update above noise (the jointly top-20% entries), averaged over all 196 weight matrices and over every teacher pair in the library: 6 pairs at K=4, 120 at K=16, 496 at K=32. The rate is 47.6%, 45.6% and 45.1% respectively, so the conflict is a property of the adapters rather than of how many you happen to have. Subspace affinity projects each student's total update onto the span of the teacher deltas; the rank sweep is K=4.

D THE SEMI-REAL LIBRARY IN FULL

Every cell of the six-skill library — both community adapters, our four contracts and MMLU — for every arm, with all three OPD seeds.

The learned-router row is a single seed against the three of the row above it, so it is worth reading cell by cell. It lands inside the three-seed spread on `bare-code` (100.0 against 96.7 ± 5.8) and `json` (20.0 against 20.3 ± 2.4), a little under it on the other four — closest on `sql`, 95.0 against 96.7 ± 0.8, and furthest on `kabyle`, where it matches the community teacher's own 70 instead of clearing it — and above it on MMLU, 60.0 against 57.5 ± 1.4. One seed against three, and the direction of the result does not change.

Table 3: The K=32 library skill by skill. †marks the three skills whose own teachers stay below 20 even after direct fine-tuning.

skill	teacher	TA	TIES	off-p.	OPD
`math`	46.5	0.0	0.0	40.6	48.9
`code`	43.5	0.0	0.0	38.9	44.4
`json`	18.5	0.0	0.0	16.7	20.3
`id`	83.5	0.0	0.0	72.2	79.4
`sum`	56.5	0.0	0.0	23.6	54.2
`de2en`	74.5	0.0	60.8	72.8	75.8
`sql`	98.0	0.0	0.0	96.7	95.3
`sent`	96.0	0.0	3.3	92.8	89.2
`mcq_xml`	87.0	0.0	0.0	84.4	85.8
`boolq`	82.5	0.0	78.3	82.2	78.3
`es2en`	83.0	0.0	77.5	78.3	77.2
`zh2en`	81.5	0.0	75.8	81.1	82.8
`arith`†	11.0	0.0	0.0	1.1	6.1
`dow`†	16.0	0.0	7.5	16.4	18.9
`sortnum`	96.5	0.0	1.7	61.1	73.1
`roman`	99.0	0.0	25.0	87.5	69.2
`fr2en`	80.8	0.0	76.7	82.5	83.9
`ru2en`	78.3	0.0	68.3	75.0	74.7
`ar2en`	73.3	0.0	61.7	68.6	72.5
`pt2en`	80.0	0.0	70.8	78.1	78.1
`capital`	100.0	0.0	0.0	100.0	100.0
`element`	100.0	0.0	9.2	100.0	100.0
`morse`	100.0	0.0	0.0	91.1	28.9
`toroman`	99.2	0.0	0.0	71.7	69.4
`tblmd`	100.0	0.0	0.0	100.0	100.0
`hex2dec`†	3.3	0.0	0.0	0.6	1.1
`phone`	100.0	0.0	0.0	100.0	98.6
`time24`	100.0	0.0	0.0	100.0	100.0
`acronym`	100.0	0.0	0.0	93.1	96.9
`isodate`	100.0	0.0	0.8	100.0	100.0
`entities`	100.0	0.0	0.0	100.0	100.0
`langid10`	94.2	0.0	0.8	92.8	90.8
mean, 32 skills	77.6	0.0	19.3	71.9	71.7
mean, 29 skills†	84.6	0.0	21.1	78.7	78.2

Table 4: A library of two real community adapters and four of our contracts, on Qwen2.5-1.5B. The base row is measured, not assumed: five of the six contracts are out of reach for it, but `sql` is not, which is why the merge driving that column to 0.0 is a loss rather than a wash. OPD matches or exceeds what either third-party adapter achieves alone, in all three seeds; the MMLU cell of the *each adapter alone* row averages the six adapters measured individually.

	real community adapters		*our contracts*				
K=6 library	kabyle	bare-code	math	json	id	sql	MMLU
base model (no adapter)	0.0	0.0	0.0	0.0	0.0	73.3	57.5
each adapter alone	70.0	90.0	46.5	18.5	83.5	98.0	55.8 $\pm$2.0
TA merge	15.0	70.0	0.0	8.3	47.5	0.0	37.5
OPD (3 seeds)	81.7 $\pm$7.6	96.7 $\pm$5.8	51.4 $\pm$2.5	20.3 $\pm$2.4	83.6 $\pm$1.3	96.7 $\pm$0.8	57.5 $\pm$1.4
OPD, learned router	70.0	100.0	45.0	20.0	81.7	95.0	60.0

E Extended related work

Weight-space composition runs from averaging (Wortsman et al., 2022) through task arithmetic (Ilharco et al., 2023), interference-aware trimming (Yadav et al., 2023; Yu et al., 2024), and data-informed weightings (Matena & Raffel, 2022; Jin et al., 2023); it is usually evaluated at small K on tasks where the base retains zero-shot ability — the cushion our contract setting removes. Casting

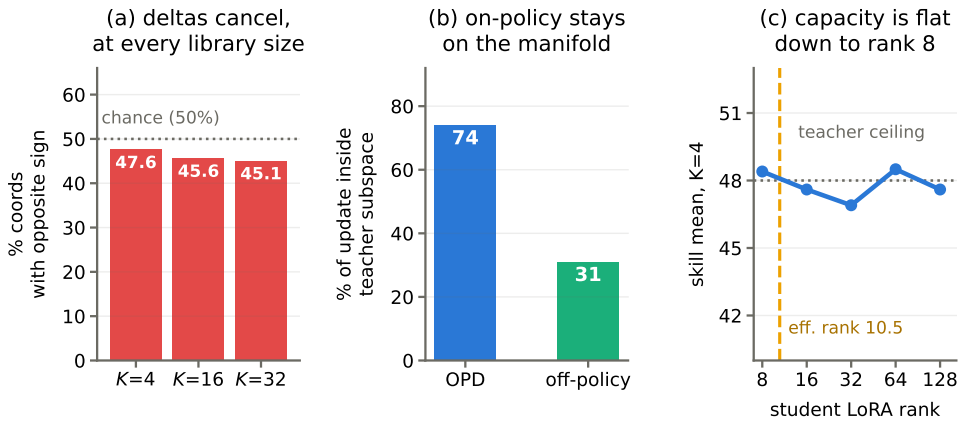


Figure 7: Why the two routes diverge. (a) Teacher updates disagree in sign almost as often as random vectors would, at every library size we measured, so summing them cancels. (b) The update OPD learns stays inside the teachers' subspace; the one imitation learns does not. (c) The consolidated update is low-rank, and the student's quality is flat down to rank 8.

the merge itself as multi-teacher distillation on scarce unlabeled data is also established (Dalili & Mahdavi, 2025). In language models a single student can absorb four soft-prompt teachers when each example is routed to its own task's teacher (Ma et al., 2026b), which presumes the task of every example is known; our router is what removes that presumption. On the distillation side, classic KD matches teacher distributions off-policy (Hinton et al., 2015; Kim & Rush, 2016) while GKD (Agarwal et al., 2024) and MiniLLM (Gu et al., 2024) train on student-generated sequences; we use that machinery as a *library operator* and find retention, not accuracy, to be where imitation fails. Sequential arrival relates to forgetting regularizers (Kirkpatrick et al., 2017), which constrain the weights toward the past; our probe instead asks what the current policy can already reach and picks a strategy from the answer.

Why the student sometimes passes the teacher. A distilled student should approach its teacher and stop there (Wang et al., 2026a), yet three of our numbers sit above the diagonal: OPD at $K{=}2$ and $K{=}4$ in Figure 1, and both real community adapters in Table 4. Three effects account for it, and none of them is the operator exceeding what it was taught. First, the excess is small against grader noise: teacher diagonals are measured on the same 120–200 held-out items as the student, so a one-to-three-point gap is inside the interval a paired bootstrap puts on either score. Second, the student sees traffic no teacher ever saw. One fifth of every batch is anchor data distilled from the frozen base, and on the real library that is enough to repair a teacher's own defects: the Kabyle adapter degenerates on prompts outside its dialogue distribution, and a student that has also been taught the base's behavior on those prompts does not. Third, the student is initialized from a merge of all K adapters, so it starts with directions no single teacher has; Section 5.1 shows this is what lets math at $K{=}4$ finish above `t_math`. We therefore read the ceiling as soft.

The closest published neighbour of our contract-competition failure is the style bias observed when one student is distilled from teachers that write differently (Wu et al., 2026b); there the conflict is stylistic and ours is over which contract owns an input distribution, but both are decided by training dynamics rather than by routing.

F Scope and Further Limitations

Some limits we can only name, not measure. Our contracts are behaviors with deterministic graders, and we have not tested adapters carrying knowledge instead of style; everything here runs on 1.5B to 8B models; and imitation is not useless, since with soft targets it closes most of the greedy-score gap (Section H). Its failure surface is retention, distribution quality and reasoning transfer, which happens to be where a consolidation operator cannot afford to fail.

G CONTRACT SKILLS AND GRADERS

The $K{=}4$ core: bare GSM8K question → final integer only; bare MBPP intent → function body (executed against unit tests); bare NQ question → one-line JSON {"answer": ...} (exact match on the value); English sentence → Indonesian translation (language-ID + chrF joint). The $K{=}32$ extension adds 12 skills at $K{=}16$ (summarization-style, three translation directions, SQL, sentiment tag, XML-wrapped MCQ, boolean QA, date/number formats) and 16 map/transform skills at $K{=}32$ (fr/ru/ar/pt→en, capital cities, chemical elements, Morse, Roman numerals, table→markdown, hex→dec, phone normalization, 24h time, acronym expansion, ISO dates, entity extraction, 10-way language ID) — each verified SFT-learnable before inclusion except the three pre-registered artifacts (`arith`, `dow` teacher diagonals 11/16 at $K{=}16$ design time; `hex2dec` 3.3), which two independent 1.5–1.7B models fail to learn even via direct SFT. Composite suites (`gsm-id`, `gsm-json`, `trans-json`) present instructions the base can follow zero-shot, testing contract–instruction interaction. All graders are deterministic and unit-tested; the two real-adapter graders (Section 5.5) were reverse-engineered from behavior samples.

H WHY ON-POLICY: FULL CONTROLS

Table 5: Consolidating four adapters, scored on the complete test sets rather than samples ($n{=}3819$). This is the table Section 5.1 quotes.

	GSM8K	MBPP	NQ-JSON	ID-trans	mean
base model (no adapter)	0.0	42.6	0.0	0.0	10.7
TA merge	0.0	35.4	15.2	80.1	32.7
off-policy KD	33.8	34.2	17.0	77.7	40.7
OPD (oracle router)	50.0	41.4	17.2	81.7	47.6
OPD (centroid router)	48.9	41.4	17.1	80.3	46.9

Paired per-item bootstrap at $K{=}32$, OPD versus off-policy KD, over the 120 held-out items of each suite (10,000 resamples, 95% CI). Six suites separate in our favour and three against; the other twenty-three do not separate, and pooled over all 3,840 items the two are indistinguishable ($+0.1\,[-1.1, +1.2]$). Ours: summarization $+30.8\,[+20.8, +40.8]$, list sorting $+25.0\,[+16.7, +33.3]$, math $+12.5\,[+2.5, +22.5]$, Indonesian $+7.5\,[+0.8, +14.2]$, Chinese→English $+3.3\,[+0.8, +6.7]$, arithmetic $+3.3\,[+0.8, +6.7]$. Theirs: Morse $-60.0\,[-69.2, -50.0]$, Roman numerals $-19.2\,[-29.2, -10.0]$, sentiment $-5.8\,[-10.8, -1.7]$. The split is close to clean: the two suites imitation wins by a wide margin are character substitutions it can copy verbatim, and the six we win all require a procedure the copy does not carry. Sentiment is the exception in its column, and a narrow one — 5.8 points on a three-label classification.

Paired per-item bootstrap at $K{=}4$ on the full test sets (10,000 resamples, 95% CI). OPD versus TA merge: math $+50.0\,[+47.3, +52.6]$, code $+6.0\,[+2.6, +9.4]$, json $+2.0\,[+0.4, +3.7]$, id $+1.6\,[-0.1, +3.3]$. OPD versus off-policy KD: math $+16.1\,[+13.4, +18.9]$, code $+7.2\,[+3.2, +11.2]$, id $+4.0\,[+2.0, +6.0]$, json $+0.2\,[-1.3, +1.7]$ (a knowledge-limited suite where the two tie). Factorization at $K{=}4$ (skill means): off-policy hard-CE 41.8; off-policy soft-KL 46.4; on-policy forward-KL 49.1; on-policy reverse-KL 48.5. Wall-clock-matched off-policy (hard labels) *decays* with compute: 41.8 (400 steps) → 32.8 (1200) → 30.5 (2400, 3600) — to merge level. Soft-target off-policy does not collapse (46.4 → 48.4 at 1200 steps, matching OPD greedy) but loses on retention (MMLU 53.5 vs. 56.5) and sampled quality (pass@8 54 vs. 63); the on-policy residual advantage is precisely {distribution quality, retention}. Joint multi-task SFT with all original data reaches a mean of 47.1.

I DISTRIBUTION QUALITY (PASS@k)

Math, 100 items, 8 samples at $T{=}1$: teacher 67, OPD 63, OPD+router 63, off-policy 30. Off-policy greedy EM 34.5 vs. sampled pass@1 5.6 (brittle argmax ridge); OPD 21.6. At $K{=}16$: student pass@8 66 vs. teacher 67 — no reverse-KL diversity collapse at any tested K.

J ROUTING ACCURACY, SKILL BY SKILL

Held-out routing accuracy for the two prompt routers of Section 5.3, measured on the 300:420 slice of each skill's prompt pool, which neither router saw while it was built. Both use the same base-model embeddings and the same 200 example prompts per skill; they differ only in the decision rule.

Table 6: Routing accuracy skill by skill at $K=32$: centroid nearest-neighbour against the learned multinomial router, on the same prompts and the same features.

skill	centroid	pklogit	skill	centroid	pklogit
`math`	95.8	100.0	`ru2en`	88.3	93.3
`code`	100.0	100.0	`ar2en`	97.5	100.0
`json`	87.5	97.5	`pt2en`	95.0	96.7
`id`	94.2	99.2	`capital`	97.5	100.0
`sum`	90.0	100.0	`element`	100.0	100.0
`de2en`	86.7	93.3	`morse`	100.0	100.0
`sql`	100.0	100.0	`toroman`	100.0	100.0
`sent`	63.3	100.0	`tblmd`	100.0	100.0
`mcq_xml`	95.8	100.0	`hex2dec`	100.0	100.0
`boolq`	80.8	99.2	`phone`	98.3	100.0
`es2en`	85.8	99.2	`time24`	100.0	100.0
`zh2en`	69.2	98.3	`acronym`	100.0	100.0
`arith`	100.0	100.0	`isodate`	100.0	100.0
`dow`	100.0	100.0	`entities`	100.0	100.0
`sortnum`	100.0	100.0	`langid10`	14.2	89.2
`roman`	93.3	97.5	`general`	86.7	95.8
`fr2en`	93.3	98.3			
overall, all 33 skills	**91.3**	**98.7**			

The discriminative rule helps most exactly where the centroid rule is weakest, and the skills it rescues are the ones whose prompts share a surface form with another skill's: `langid10` against the seven translation teachers, `sent` and `boolq` against each other and against the general anchor pool. On skills whose prompts are already separable, the two rules agree to within a point.

K HOW MUCH PROVENANCE THE ROUTER NEEDS

The learned router is fitted on labelled example prompts, which is the one piece of provenance our setting does assume, so its size is a result, not a detail. We refit it at six budgets, three seeds each, and score every fit on the same held-out prompts (each skill's evaluation split plus 120 generic prompts, 3,960 in all). No student is trained: the router is a multinomial logistic regression on frozen prompt embeddings, so a budget costs one embedding pass.

Table 7: Routing accuracy against the number of labelled example prompts per adapter, at $K=32$, over three seeds.

examples per adapter	labelled prompts	centroid NN held-out routing accuracy (%)	pklogit (ours) held-out routing accuracy (%)
5	165	90.5 $\pm$2.6	95.1 $\pm$0.7
10	330	89.7 $\pm$0.1	96.5 $\pm$0.5
20	660	91.2 $\pm$0.9	97.8 $\pm$0.2
50	1,650	90.8 $\pm$1.4	98.5 $\pm$0.1
100	3,300	91.6 $\pm$0.2	98.9 $\pm$0.1
200	6,600	91.5 $\pm$0.1	99.0 $\pm$0.1

The provenance the recipe needs is small and cheap — five prompts an adapter, the kind of thing a model card usually carries anyway — so the gap between "no metadata" and what we actually use is 165 labelled prompts, not a routing table. And the centroid rule is flat in budget while the discriminative rule is not, which says the failure in Figure 4a is a property of mean-centroid geometry that more examples cannot fix. At the operating point the paper uses, 200 per adapter, this protocol puts pklogit at 99.0 and centroids at 91.5, within 0.3 of the 98.7 and 91.3 that Table 6 reports under the run's own held-out split.

L Out-of-Domain Abstention

A deployed consolidated adapter also receives prompts belonging to none of its contracts, and the router of Section 5.3 will happily assign them to a teacher. For completeness we report an abstention mechanism, not used in any headline result. The prompt router extended with a P5 self-similarity threshold abstains on OOD prompts: in-domain false-abstain 1–7%; OOD catch rates 97/85/41/2% across four held-out families in v1; adding a length z-score feature fixes the long-text miss (sum-type: 2% → 98%), yielding 98/99/100/63% at 1–10% false-abstain. Online centroid EMA updates give no further gain (20 exemplars already at 96.9% of ceiling accuracy).

M Safety, Instruction Following, and Inference-Time Routing

With a refusal-contract teacher in the library (Qwen2.5-1.5B): AdvBench refusal — base 100%, TA merge **0.8%**, OPD 100%; benign over-refusal (teacher 78% answer rate) repaired to 94.5% by OPD. IFEval-lite, our reimplementation of the verifiable-instruction protocol of Zhou et al. (2023) (364 instructions, 17 checkers): base 38.2 > OPD 36.5 > off-policy 28.3 > merge 20.9. Inference-time adapter routing (one adapter per request) matches teachers on single skills but scores 0.5 on `trans-json` composition vs. OPD 54: composition requires the behaviors in *one* set of weights.

N Merge Hyperparameter Sweeps and Exact-Delta Corrections

TA λ-sweep at $K{=}4$: best 38.2 at $\lambda{=}0.5$; each λ zeroes a different contract (no Pareto point). Materializing exact full-ΔW merges (no SVD truncation): at $K{=}4$, exact TIES 34.4 (its SVD form reports 30.9) and exact DARE 34.0 (33.1) against plain summation's 33.6, which is exactly low-rank and therefore unaffected; at $K{=}8$, exact TIES reaches 22.3 where plain summation scores 4.0; at $K{=}32$, exact TIES reaches 19.3 against the base model's 19.7 and plain summation's 0.0. Every merge number in the paper is the exact one, and Figure 1 plots $\max$(TA, TA-λ, exact TIES) at each K; conclusions unchanged. SVD approximation errors of trimmed merges are large (TIES 0.57, DARE 0.91 relative) because trimming destroys low-rank structure; TA is exactly low-rank.

O The Reachability Probe

For a candidate contract with first token t_1: sample held-out bare prompts, compute $P(t_1 \mid \text{prompt})$ under the current student; median $\geq 10^{-6}$ predicts a 150-step increment installs the contract; below, merge-then-teach. Calibrated at 150-step budgets. The threshold transfers across families; which skills clear it does not. Three of twelve contracts change branches between Qwen2.5-1.5B and SmolLM2-1.7B: `mcq_xml` reads $5.6{\times}10^{-7}$ on Qwen and $7.3{\times}10^{-6}$ on SmolLM2, `arith` $5.2{\times}10^{-7}$ against $2.5{\times}10^{-5}$, and `dow` runs the other way, $2.3{\times}10^{-5}$ on Qwen against $2.2{\times}10^{-12}$ on SmolLM2 — seven orders of magnitude on the same contract and the same probe. The probe is released as a standalone tool alongside the code.

P Lifecycle: the full arrival sequence

Twelve adapters arrive in sequence on top of an initial $K{=}4$ student; every strategy is evaluated on all sixteen skills at the end. Cost is the exact number of distillation steps spent over the whole sequence.

Table 8: What the probe buys. The two naive rows spend the same or more compute and install fewer contracts; retraining from scratch installs everything but costs 2.7× the steps. The arrival chains are scored on 100 held-out items per skill and the two reference rows on 200, so the comparison to make is within the chain; the retrain row is a three-seed mean (59.5/58.1/59.9); the others are single runs.

strategy over the arrival sequence	16-skill mean	contracts installed	OPD steps
naive increment at every arrival	46.1	9/12	1,800
naive, then repair the failures at the end	51.2	11/12	2,250
probe-driven repair at arrival	**54.9**	**12/12**	**1,800**
the same recipe on SmolLM2-1.7B	55.8	12/12	1,800
retrain from scratch at every arrival	59.2	12/12	4,800

Table 9 is the probe-driven row, arrival by arrival: the reading the probe took on the running student, and the branch that reading selected. Reachable contracts get a 150-step increment taught by the

running student itself; unreachable ones are re-initialized from a merge that includes the newcomer and then taught for the same 150 steps. Both columns are read out of the run's own logs, so the table cannot drift from what was executed.

Table 9: The twelve arrivals in order, with the probe reading taken on the running student and the branch that reading selected. Both columns are read straight out of the run's own logs, by `scripts/arrivals_table.py` in the released code.

#	arriving contract	$P(\text{tok}_1)$	probe says	how it was taught
1	`sum`	2.6×10^{-5}	reachable	150-step increment
2	`de2en`	1.0×10^{-3}	reachable	150-step increment
3	`sql`	1.1×10^{-3}	reachable	150-step increment
4	`sent`	8.7×10^{-9}	far mode	re-merge, then 150 steps
5	`mcq_xml`	5.6×10^{-7}	far mode	re-merge, then 150 steps
6	`boolq`	1.5×10^{-13}	far mode	re-merge, then 150 steps
7	`es2en`	6.4×10^{-2}	reachable	150-step increment
8	`zh2en`	6.5×10^{-2}	reachable	150-step increment
9	`arith`	5.2×10^{-7}	far mode	re-merge, then 150 steps
10	`dow`	2.3×10^{-5}	reachable	150-step increment
11	`sortnum`	3.1×10^{-2}	reachable	150-step increment
12	`roman`	2.7×10^{-1}	reachable	150-step increment

Eight of the twelve were reachable and four were not. Three of the four the probe diverted — `sent`, `mcq_xml`, `arith` — are precisely the ones a naive increment fails to install. The fourth is `boolq`, which reads $1.5{\times}10^{-13}$ on this trajectory, twelve orders below threshold, yet on the naive run, whose student reached that point by a different route, a plain increment installed it anyway. So the probe's calls line up with what naive increments actually achieve on eleven of twelve contracts, not twelve of twelve, and its one error is conservative — it pays for a re-merge that was not needed rather than skipping one that was.

A contract counts as installed when the student emits its output format on held-out prompts at all. The metric is bimodal and needs no threshold: a contract that was never installed sits at exactly 0.0, and every installed one is at 0.69 or above. By that criterion naive increments install nine of the twelve and miss three "far-mode" contracts entirely, because reverse KL never samples them. Repairing those three at the end of the sequence recovers two, since by then the student has drifted; repairing at the moment of arrival recovers all three, which is the recipe we recommend.

Two details generalize. Reachability belongs to the trajectory rather than the skill: run the same twelve arrivals on SmolLM2-1.7B and three of the twelve change branches — `mcq_xml` and `arith` become reachable there while `dow`, reachable on Qwen at $2.3{\times}10^{-5}$, drops to $2.2{\times}10^{-12}$ and needs the merge. A fixed list of hard skills would not transfer between families; the probe does, unchanged. And the gap that remains to full retraining is slow competence transfer, not failed installation: formats appear within 150 steps while the underlying ability lags. Extending the two worst increments from 150 to 450 steps recovers most of it — sorting a list of numbers goes from 18 to 67, and writing Roman numerals from 12 to 81.

Q Seeds, Costs, and Infrastructure Notes

Wall-clock on one A100 at $K{=}32$, measured end to end: training the thirty-two teacher adapters, which every method in the paper needs, 1.7 hours; merging them, about a minute; off-policy generation plus fine-tuning, 1.6 hours; OPD, 6.2 hours. A later arrival costs 150 steps, not a re-consolidation.

Hyperparameters, identical across arms unless noted: student LoRA rank 64 ($\alpha{=}64$, dropout 0) on all attention and MLP projections; AdamW at $\text{lr} = 10^{-4}$, $\beta = (0.9, 0.95)$, 10 warmup steps, no weight decay; 400 steps at $K{\leq}16$ and 500 at $K{=}32$; 24–48 prompts per step, rollouts sampled at temperature 1.0 and capped at 256 new tokens; anchor fraction 0.2; micro-batch 1–4 by model size with gradient checkpointing. Teachers are rank-16 ($\alpha{=}32$) SFT adapters, 2 epochs (4 for code). Off-policy KD uses the same student rank and step count on 1,000–2,000 generated samples per teacher. Each campaign is one script under `scripts/`: `20_campaign11.sh` is the $K{=}32$ library, `28_campaign13_l8.sh` the 8B replication, `26_k32_pklogit.sh` the discriminative router, `35_hidose30.sh` the dose–response arm. All headline arms are 3–5 seeds ($K{=}4$ OPD: 48.2 ± 0.6, 5 seeds; promptknn 47.9 ± 1.0; $K{=}16$ 59.2 ± 0.9; 4B 63.3 ± 1.2; real-library 81.7 ± 7.6

/ 96.7 ± 5.8; pklogit K=32 langid10 90.3 ± 0.5). Measured single-A100 wall-clock at K=32 in Section 5.5. Infrastructure notes for reproduction: third-party weight downloads must be integrity-verified before entering a pipeline (we encountered a hub client silently returning a checkpoint missing a 4.9 GB shard); progress monitoring should combine file mtimes, process liveness, step counts, and elapsed-time consistency — step-count spot checks alone produced a false "stall" diagnosis that survived until elapsed-gap analysis falsified it.

R WHAT THE ADAPTER ECOLOGY ACTUALLY LOOKS LIKE

The audit behind Section 5.5. We searched the hub for adapters declaring Qwen2.5-1.5B-Instruct as their base, kept those that load as valid PEFT checkpoints, and then asked of each whether it produces a behavior distinguishable from the base that a deterministic grader could score.

Table 10: The adapter-ecology funnel: how many adapters survived each stage of the hub audit, and what removed the ones that did not.

stage	count	what removes them
declared for this base model	31	the starting pool
loads as a valid PEFT checkpoint	15	wrong base, corrupt shards, missing config
behaviorally distinct from the base	6	output indistinguishable from π_0
gradable behavioral contract	**2**	mislabeled or degenerate

The four that were distinct but ungradable are instructive. One is advertised as a German PII redactor and, given German text containing names, emails and phone numbers, repeats them verbatim. One is an authorship-style adapter that degenerates into repetition on any non-English prompt. The remaining two produce a recognizable style with no property a grader can check. Behavioral divergence from the base is easy to measure and does not imply the adapter does what its card says.

S CONTRACT-COMPETITION DOSE–RESPONSE

`trans-json` (real library, pklogit routing) under compositional-anchor share of total teaching traffic: 0% $\rightarrow$ mean 47.1 (seeds 77.5/16.7); 4% $\rightarrow$ 28.3 ± 20.5 (dilution); 15% $\rightarrow$ 59.4 ± 30.6 (76.7/77.5/24.2); 30% $\rightarrow$ 67.9 (55.8/80.0), with gsm-style composition rising monotonically to $5\times$ the oracle arm (gsm-json 30.8/40.0 vs. 8.0). Contract-priority sampling at K=32 (`json` $4\times$): installs in 2/2 seeds (19.2/18.3) vs. 3/5 unweighted; best overall arm (79.5). Neither lever reaches determinism at tested settings.