THE DELTA IS NOT THE SKILL: ZERO-PRIOR ELICITATION TRANSFERS LORA BEHAVIOR ACROSS MODEL FAMILIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Reusing low-rank adapters (LoRA) across model sizes, generations, and architectural families is increasingly essential for deploying specialized capabilities. However, native retraining requires inaccessible training data, while existing synthetic-data transfers rely on task-specific priors (task names, seed examples, labels, or auxiliary discriminators). Weight-space transfer avoids these priors, but we uncover a sharp dichotomy: within a shared parameter basis, zero-edit adapter transplantation is near-lossless (recovering 92–116% of native gains), whereas across distinct families, weight-space routes collapse ($\mathrm{TRR} \leq 0.08$) because low-rank deltas are base-relative. What survives across model boundaries is behavior, not weights. We introduce *zero-prior skill elicitation*, a framework that extracts adapter capabilities without task identities, seed data, labels, discriminators, or white-box access. By identifying inputs where adapter and base generations diverge, evolving these probes to broaden coverage, and distilling the adapter-equipped teacher's completions into students, we transfer complex skills purely through text. Distilling from a single 397B MoE teacher LoRA matches or exceeds native supervised fine-tuning across four student models in two architectural families, reducing target training compute by 2.6–$3.0\times$. Gains are pronounced on multi-step reasoning (elevating 9B MATH-500 by $+27.2$ points; $\mathrm{TRR} = 3.39$), competition mathematics, and out-of-distribution instruction following, with zero verbatim n-gram benchmark leakage and successful generalization to Granite models.

1 INTRODUCTION

Low-rank adaptation (Hu et al., 2021) and related parameter-efficient methods (Houlsby et al., 2019; Dettmers et al., 2023; Liu et al., 2024b) have made task skills portable in principle: a few hundred megabytes of adapter weights encode what a fine-tuning run taught a model, and serving systems already host thousands of adapters on a shared base (Sheng et al., 2023; Chen et al., 2024). Production deployments increasingly need that portability to be real—a fleet of models spanning several sizes, generations, and families should be upgradable from one maintained adapter. This paper targets exactly that *train-once-deploy-many* setting: given a single teacher-side LoRA, produce working students across model sizes, generations, and families. The obstacles are concrete. ❶ **Per-Target Retraining**: the default answer, fine-tuning every target natively, requires the original labeled data—often unavailable for privacy, licensing, or organizational reasons—and a full training run for each new deployment.

Existing transfer methods avoid retraining but presuppose the priors the adapter was supposed to abstract away. ❷ **Task-Prior Dependence**: synthetic-data pipelines such as Trans-LoRA (Wang et al., 2024) and TiTok (Jung & Kim, 2026) regenerate training data for the target, but their generation step is *task-conditioned*: it starts from the task's identity, per-task templates, retained seed examples, and (for Trans-LoRA) labels plus a separately trained discriminator. The technical root is that these methods use the base-vs-adapted contrast only as a *filter* over data they already know how to generate; remove the task identity and there is nothing left to filter.

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.

Weight-space transport methods avoid data priors as well, but rest on an assumption that we show is load-bearing. ❸ **Basis Dependence**: SVD-subspace and activation-Procrustes projections (Farhadzadeh et al., 2025; Xia et al., 2025; Rinaldi et al., 2026) report gains mainly on diffusion models or multiple-choice benchmarks; on executable generation tasks we measure a sharp dichotomy (§4). Between checkpoints sharing a parameter basis, transplanting the adapter with *zero edits* is near-lossless (transfer-rate ratio, TRR: the fraction of the native-training gain that the transferred adapter recovers, formalized in §3; here 0.92–1.16), and an explicit gauge map adds nothing. This agrees with recent delta-transfer results across versions of one model family (Huang et al., 2024; Khan et al., 2025; Lin et al., 2025; Cao et al., 2025; Li et al., 2026). Across families, every weight-space route we test—SVD alignment, behavioral deltas, trust-region field edits, even a learned adapter compiler—collapses to $\mathrm{TRR} \leq 0.08$. The technical reason is that the LoRA delta is *base-relative*: it is meaningful only in the parameter coordinate system it was trained in, so no skill-independent coordinate map can carry it across families.

Since only behavior travels across families, we ask whether the adapter alone provides enough information to recover it. Our procedure (Figure 2) is simple: contrast the teacher's generations with and without the adapter on a generic probe pool, score the divergence, and evolve probes toward inputs where the adapter changes behavior most. The adapter-equipped teacher then answers the discovered probes, and students are trained on the resulting trajectories. We call this procedure *zero-prior skill elicitation* because it needs no task name, no original data or seed examples, no labels, no discriminator, and no white-box access. The design responds to the three obstacles directly: transferring behavior rather than weights sidesteps ❸; discovering probes from the divergence signal alone removes ❷; and because the elicited trajectories are plain text, one teacher-side elicitation is paid once and distills into any number of students, addressing ❶. Related work has established weaker versions of this observation: model diffing shows that fine-tunes are readable (Minder et al., 2026b; Brzozowski et al., 2026), TiTok uses the base-vs-adapted contrast as a data filter (Jung & Kim, 2026), and proxy-tuning and emulated fine-tuning apply a tuned-minus-untuned logit difference at decoding time, which requires a shared vocabulary and keeps the source models in the inference loop (Liu et al., 2024a; Mitchell et al., 2023). What has been missing is evidence that the divergence signal by itself can produce a working student in a new model. Providing that evidence is the main contribution of this paper.

We evaluate zero-prior skill elicitation by transferring a production-scale multi-domain LoRA from a 397B MoE teacher to four students from 4B to 27B in two families (Qwen and Llama), and test a third family (Granite). Across extensive evaluations spanning mathematical reasoning, competition problems, science, and instruction following, our distilled students consistently match or outperform models trained natively on original labeled data. Without receiving any task names, human labels, or retained examples, the elicited trajectories unlock substantial gains on complex reasoning tasks where weight-space transfer collapses, while reducing target-side deployment costs by 2.6–$3.0\times$.

Our contributions are:

- We formulate cross-family LoRA reuse as *zero-prior skill elicitation*: recovering the adapter's behavior distribution from the base-vs-adapter divergence alone, motivated by a measured dichotomy showing that within a shared parameter basis zero-edit transplant is near-lossless (TRR 0.92–1.16) while every cross-family weight-space route collapses ($\mathrm{TRR} \leq 0.08$).
- We propose a train-once-deploy-many pipeline that couples divergence scoring, coverage-driven probe evolution, and distillation, requiring no task identity, original data, labels, discriminator, or white-box access (Table 1 positions prior work along these axes).
- We evaluate the pipeline with a 397B MoE teacher across four student models in two families. Zero-prior distillation consistently outperforms native supervised fine-tuning across all student scales on aggregate mean scores (maintaining a $+6.0$ to $+13.9$ point advantage) as well as on MATH-500 and IFEval at every size, while reducing target training cost by 2.6–$3.0\times$. The zero-prior probes also transfer to a third family (Granite-3.3-8B, §6).
- We rule out alternative explanations with an unadapted-teacher control, token-matched generic SFT, a contamination audit, and multi-seed replicas. In a small-scale study, zero-prior probes match or exceed distillation supervised by the original task questions under matched token budgets ($\mathrm{TRR} = 0.85$).

2 RELATED WORK

Adapter transfer with synthetic data. Trans-LoRA (Wang et al., 2024) generates target data, labels it with a source expert, and filters it with a trained discriminator, requiring task identities, labels, and retained seed examples. TiTok (Jung & Kim, 2026) replaces the discriminator with contrastive loss scoring, but still conditions generation on task names and few-shot seeds. These pipelines use contrast only as a *filter* over pre-existing task distributions; removing the task prior halts data generation entirely. In contrast, our approach uses behavioral divergence as an open-ended *discovery* objective over neutral inputs. Hypernetworks (Charakorn et al., 2025; Liang et al., 2026) generate weights from task descriptions, but remain tied to explicit task conditioning and a single base architecture.

Weight-space and decoding-time transfer. Subspace projection (LoRA-X, Farhadzadeh et al., 2025; Cross-LoRA, Xia et al., 2025) and activation Procrustes transport (Theseus, Rinaldi et al., 2026) attempt direct weight transformation. Within a single model family, transferring weight deltas preserves capabilities across checkpoint updates (Huang et al., 2024; Khan et al., 2025; Lin et al., 2025; Cao et al., 2025; Li et al., 2026), as do task arithmetic and model merging (Ilharco et al., 2022; Yadav et al., 2023; Yu et al., 2024; Ainsworth et al., 2022). Across heterogeneous families, however, we show that weight-space routes collapse ($\mathrm{TRR} \leq 0.08$) because low-rank deltas are coordinate-dependent and base-relative. Decoding-time methods such as proxy-tuning (Liu et al., 2024a) and emulated fine-tuning (Mitchell et al., 2023) require shared vocabularies and retain source models during inference, precluding standalone student deployment.

Model diffing and capability elicitation. Activation lenses (Minder et al., 2026b;a) and contrastive decoding diffing (Brzozowski et al., 2026) inspect fine-tuned updates under white-box access, while fine-tuned models can partially self-report behavioral changes (Betley et al., 2025). While these methods demonstrate that fine-tuning deltas are *readable*, they do not transfer capabilities into new models. We demonstrate the stronger operational property: black-box behavioral divergence is *sufficient* to discover curricula and recompile capabilities into independent student models.

Distillation and prompt evolution. Standard knowledge distillation (Hinton et al., 2015; Kim & Rush, 2016) and instruction generation (Wang et al., 2023; Taori et al., 2023) assume access to task prompts or seed distributions. Data-free distillation requires white-box gradients (Chen et al., 2019; Yin et al., 2020), while black-box model extraction (Tramèr et al., 2016; Krishna et al., 2019) primarily transfers surface stylistic traits (Gudibande et al., 2023). Prompt synthesis frameworks like Evol-Instruct (Xu et al., 2024), Magpie (Xu et al., 2025), and Rainbow Teaming (Samvelyan et al., 2024) employ mutation and coverage search across full models. We adapt evolutionary operators specifically to isolate the adapter's behavioral delta relative to its base, ensuring the elicited curriculum targets new procedural capabilities rather than pre-existing base knowledge (§7).

3 PROBLEM SETUP AND THE ZERO-PRIOR DEFINITION

Transfer-rate ratio. Let B_t be a target base model, O_t an *oracle* adapter trained natively on the target with the original labeled data, and A_t the transferred artifact under evaluation. For a benchmark with accuracy $\mathrm{acc}(\cdot)$,

$$\mathrm{TRR} = \frac{\mathrm{acc}(B_t + A_t) - \mathrm{acc}(B_t)}{\mathrm{acc}(B_t + O_t) - \mathrm{acc}(B_t)}. \tag{1}$$

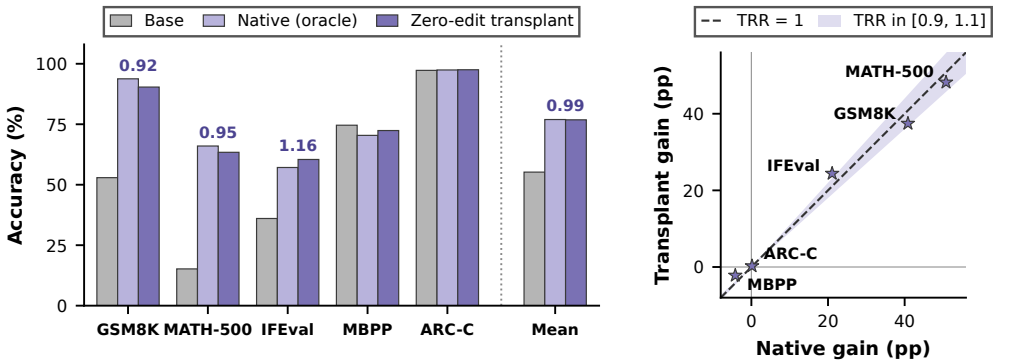
$\mathrm{TRR} = 1$ means transfer recovers the full native-training gain; TRR can exceed 1 when the transferred skill (e.g., from a much larger teacher) surpasses native training. Where the oracle underperforms the base (negative denominator), we report absolute gains instead.

Zero-prior, operationally. We call a transfer procedure *zero-prior* if it uses none of: (i) task identity (names, descriptions, templates), (ii) original training data or retained seed examples, (iii) labels or reference answers, (iv) auxiliary task models (discriminators, verifiers), (v) white-box internals of the teacher beyond the ability to sample with and without the adapter. Table 1 positions prior work along these axes; ours is the only all-"no" row that produces a working student.

Models and tasks. The teacher is Qwen3.5-397B-A17B (MoE) carrying a mix LoRA trained once on five domains (531 steps, 5,689 labeled examples; this is the production asset whose skill we transfer—its training cost is not part of transfer cost, just as native training does not count data con-

Table 1: Operational definition of *zero-prior* transfer. Ours is the only method requiring none of the five priors while still producing a working student model. †ADL/CDD are auditing methods: they read the fine-tune but do not transfer the skill.

Method	Task id.	Data/seeds	Labels	Discrim./verif.	White-box	Student?
Trans-LoRA	yes	5 seeds	yes	yes	no	yes
TiTok	yes	5 seeds	no	no	no	yes
Theseus	per-task α	calib. acts.	no	no	yes	yes
Cross-LoRA / LoRA-X	no	no	no	no	yes	yes
ADL†	no	no	no	no	yes	no
CDD†	no	no	no	no	no	no
Ours	**no**	**no**	**no**	**no**	**no**	**yes**



(a) Accuracy on Qwen3.6-27B

(b) Gain recovered over Qwen3.6 base

Figure 1: **Cross-generation transplant** (Qwen3.5-27B LoRA → Qwen3.6-27B, greedy). (a) Accuracy of the Qwen3.6 base, the Qwen3.6-native oracle, and the Qwen3.5 LoRA attached with zero edits; the number above each group is TRR, omitted where the native gain is negative (MBPP) or negligible (ARC-C). (b) Gain over the Qwen3.6 base from transplant versus native training; points on the diagonal recover the full native gain, and the shaded band marks $\mathrm{TRR} \in [0.9, 1.1]$. Full numbers, including the Qwen3.5 source arms, are in Table 4 (Appendix A).

struction). Distillation students: Qwen3.5-27B/9B/4B and Llama-3.1-8B (Grattafiori et al., 2024), plus Granite-3.3-8B-Instruct (Granite Team, 2024) as a third family (§6). Qwen3.6-27B (same shapes, next generation) is the target of the weight transplant in §4.1. Benchmarks: GSM8K (Cobbe et al., 2021) (1319), MATH-500 (Hendrycks et al., 2021) (500), MBPP (Austin et al., 2021) (500), ARC-Challenge (Clark et al., 2018) (1172), IFEval (Zhou et al., 2023) (541), all full test sets, greedy decoding, fixed prompt protocols; IFBench (Pyatkin et al., 2026) (300, strict prompt-level, official checker) as an OOD constraint-following stressor; AIME 2024+2025 (60) (Codeforces, 2024) and HMMT (30) as harder math contest sets, scored by integer match in $\{0, \ldots, 999\}$ after the same `\boxed{}` completion prompt as MATH-500 (generation budget 2048 tokens); GPQA-Diamond (198) (Rein et al., 2023) as a harder science probe.

4 When do weights travel? A measured dichotomy

4.1 Same parameter basis: zero-edit transplant is the right answer

We first establish the positive half of the dichotomy: within a shared parameter basis, the adapter itself is the right transfer artifact. Qwen3.5-27B and Qwen3.6-27B share all 1,199 tensor names and shapes. We transplant the 3.5-native mix LoRA onto 3.6 with *zero edits* and compare against (a) the 3.6-native oracle and (b) an explicit gauge map: per-interface orthogonal Procrustes rotations Q plus a scalar gain s, fitted on task-agnostic text (24K and 200K tokens).

Figure 1 shows that transfer is essentially free within a parameter basis. The transplanted adapter lands on the diagonal wherever native training helps: it recovers 92% and 95% of the native gain on GSM8K and MATH-500 and exceeds it on IFEval, so its five-benchmark mean (76.83) is within 0.1 points of the oracle (76.95). On MBPP, where native training lowers accuracy, the transplant loses less than the oracle does. The explicit gauge map *does not help*: on GPQA the formula variant scores 41.9–42.9% against zero-edit's 48.5% (Appendix A). Diagnostics explain why: the fitted Q is nearly the identity in small interface subspaces ($\|Q - I\|/\sqrt{2d} = 0.065$ at d=48) but drifts non-orthogonally in FFN space (0.748–0.859 at d=17408), and $8\times$ more calibration tokens improve the estimate while adding only ~ 1 percentage point—the residual gap is knowledge drift, not sampling noise, and no linear coordinate map recovers it.

4.2 Across families: every weight-space route collapses

The negative half of the dichotomy is that no weight-space route we test survives a change of parameter basis. Between Qwen2.5-3B and Llama-3.2-3B we tested the weight-space design space at small scale: joint functional matching of adapter outputs on wikitext and task probes, behavior-delta matching (scalar and multi-candidate), decision- and dense-field trust-region edits, and a learned adapter compiler G_ϕ supervised on five task pairs (24M parameters, training loss converged to 0.002). Every variant lands at $\mathrm{TRR} \leq 0.08$ on GSM8K (Appendix B); a capacity control that matches *target-native* hidden states reaches $\mathrm{TRR} = 0.59$, so expressiveness is not the bottleneck—the supervision signal is. The mechanism is base-relativity: $\Delta_s = \text{skill} \ominus \text{knowledge}(B_s)$ and $\Delta_t^\star = \text{skill} \ominus \text{knowledge}(B_t)$ subtract different things; the delta is not the skill; the Procrustes diagnostics above support this mechanism.

The dichotomy yields a routing law: *same basis ⇒ transplant weights, zero edits; different basis ⇒ only behavior travels*. The rest of the paper is about extracting that behavior with zero priors.

5 Zero-prior skill elicitation

Overview. The routing law of §4 dictates the pipeline shape: across families the only transferable object is the adapter's behavior distribution, so the method must (i) find where that distribution differs from the base model's, (ii) cover it, and (iii) rewrite it into a student—all without task priors. The resulting train-once-deploy-many pipeline has four stages (Figure 2). *Stage 1* trains the teacher-side mix LoRA once (this is the pre-existing production asset; its training data need not be retained). *Stage 2* elicits the adapter's behavior: divergence scoring (§5.1) locates probes where the adapter changes the teacher's behavior, and coverage-driven evolution (§5.2) expands them into a curriculum. *Stage 3* distills the adapter-equipped teacher's trajectories on the discovered probes into any number of students (§5.3). *Stage 4* evaluates each student with TRR and absolute benchmark scores. Only the teacher and the teacher-plus-adapter are ever queried, and only through sampling; teacher-side cost is paid once and amortizes over all targets.

5.1 Divergence scoring

Without task identity, data, or labels, the only signal that distinguishes the adapter from its base under our black-box constraint is a difference in generated behavior; the first module measures exactly that. Let T be the teacher base and $T{+}A$ the adapter-equipped teacher. We build a generic probe pool self-proposed by the teacher from neutral seeds (no benchmark data), and for each probe x we greedily decode $y_a \sim T{+}A(\cdot \mid x)$ and $y_b \sim T(\cdot \mid x)$ and score

$$S(x) = d_{\mathrm{Jaccard}}(y_a, y_b) + 0.1\,\big(1 - \mathrm{ratio}(y_a, y_b)\big), \tag{2}$$

where d_{Jaccard} is token-set Jaccard distance and ratio is a sequence-similarity ratio; high $S(x)$ marks inputs where the adapter changes behavior most. The advantage over prior contrast signals is what the score does *not* need: unlike TiTok's loss excess it requires no task-conditioned generation to score, and unlike activation-difference lenses it requires no white-box access—two greedy generations per probe suffice.

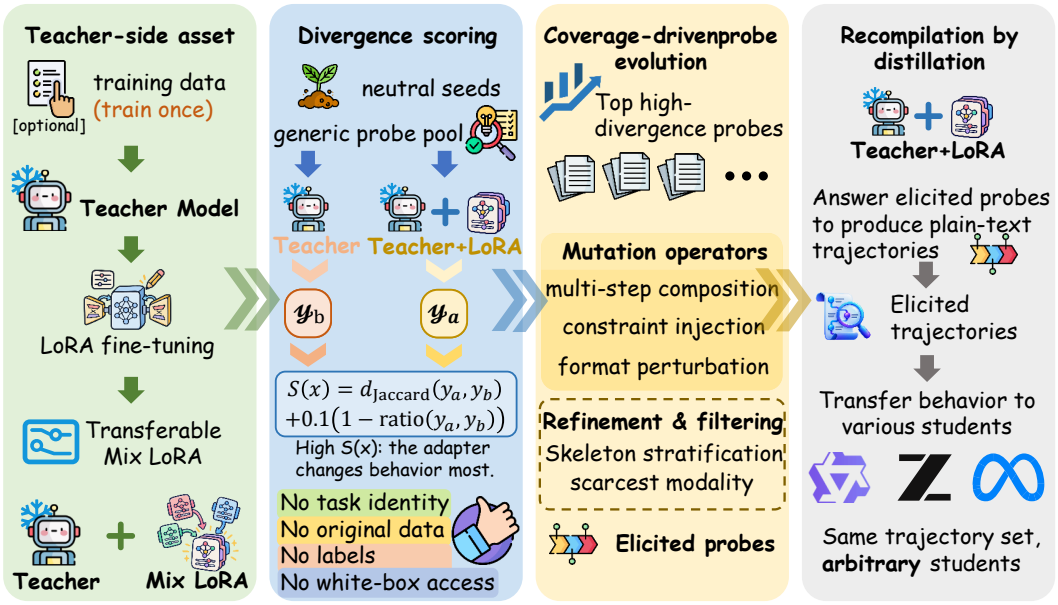


Figure 2: **Zero-prior skill elicitation.** A mix LoRA trained once on the teacher is the only transferred asset. Divergence scoring decodes each probe of a generic pool, self-proposed from neutral seeds, with and without the adapter and keeps the probes where the two outputs differ most. Coverage-driven evolution mutates these probes and refills them under skeleton stratification and scarcest-modality selection. The adapter-equipped teacher then answers the elicited probes, and the resulting plain-text trajectories train students of any family. No stage uses task identity, original data, labels, or white-box access.

5.2 Coverage-driven probe evolution

A static probe pool finds where the adapter's behavior differs but under-covers it: the pool is finite, and naive top-k selection concentrates on the divergence *mode* rather than its support. The second module turns scoring into a generator. High-divergence probes are selected as parents, mutated by generic operators (multi-step composition, constraint injection, format perturbation), and re-scored. Selection uses two constraints in one loop: stratification over answer skeletons, so the set covers the adapter's behavioral support, and scarcest-first refill over answer modalities, so code and multiple-choice holes are not left empty. This is the coverage-aware elicitation rule: 2,158 trajectories spanning 215 distinct skeletons, produced by the 397B teacher in ∼20 GPU·h with no task priors (modality histograms in Appendix D). This default set is the curriculum behind every *ours* result. Dropping the modality refill—skeleton stratification only—yields 2,071 trajectories that are more math-concentrated (∼1,100 math stems versus 759); we denote this variant *w/o coverage*. Evolution also removes the size limit of a static pool: the pre-evolution pool saturates at 1,294 usable probes, whereas evolution keeps producing new probes at no additional prior cost.

5.3 Recompilation by distillation

The final module rewrites the elicited behavior into arbitrary students. The adapter-equipped teacher answers every discovered probe; each student is then trained with a short LoRA SFT on the resulting trajectories (∼200 steps, recipe identical to the native baselines, so comparisons isolate the data). Because students consume plain text trajectories, nothing in this stage depends on the student's family, tokenizer, or width—the same trajectory set drives all four students in Table 2. Nothing in the pipeline names the tasks; notably, the elicited trajectories carry GSM-style calculator fingerprints (`<<19-18=1>>1 #### 1`)—the adapter's training traces are dug out automatically. At small scale, a zero-prior audit that removes all manually specified math priors from the pipeline autonomously re-discovers the same motif specification (Appendix C).

Table 2: **Main evaluation results.** Downstream accuracy (%) across student architectures and teacher models under greedy decoding. **Bold** marks the best result per column within each student; shaded rows denote our method. *Mean* averages the five core benchmarks common to all rows.

Student	Arm	Teacher	Math & Reasoning			Competition Math			Science	Instr. Following		Mean
			GSM8K	MATH	ARC-C	AIME24	AIME25	HMMT	GPQA-D	IFEval	IFBench	
27B	Base	—	88.25	44.40	90.22	10.00	3.33	6.67	3.03	50.09	24.67	59.53
	Native (labels)	—	91.36	62.80	95.05	6.67	6.67	13.33	**48.48**	59.70	24.33	66.65
	Cross-LoRA	—	28.43	52.80	94.89	10.00	13.33	6.67	2.53	55.08	28.33	51.91
	TiTok (no seeds)	—	88.78	62.20	94.44	6.67	3.33	3.33	47.47	61.37	36.67	68.69
	Ours	GLM-5.3-Flash	**95.53**	**74.40**	95.05	**40.00**	**36.67**	**26.67**	42.42	62.90	**42.33**	**74.04**
	Ours	Qwen3.5-397B	92.57	68.20	**97.61**	23.33	16.67	16.67	39.90	**63.59**	41.33	72.66
9B	Base	—	78.62	29.60	80.63	3.33	3.33	6.67	22.22	42.88	21.67	50.68
	Native (labels)	—	85.29	41.00	94.45	0.00	0.00	10.00	40.40	40.11	17.67	55.70
	Cross-LoRA	—	24.94	38.00	93.94	3.33	13.33	6.67	21.72	47.87	25.33	46.02
	TiTok (no seeds)	—	78.92	48.40	93.86	3.33	0.00	0.00	43.94	59.15	31.67	62.40
	Ours	GLM-5.3-Flash	**90.22**	66.20	94.45	**26.67**	13.33	**16.67**	**47.98**	42.70	23.67	63.45
	Ours	Qwen3.5-397B	88.70	**68.20**	**94.80**	23.33	**16.67**	10.00	38.38	**61.74**	**34.67**	**69.62**
4B	Base	—	20.62	43.40	79.61	6.67	13.33	13.33	5.56	29.94	19.33	38.58
	Native (labels)	—	77.94	47.00	91.81	6.67	3.33	6.67	39.39	40.48	18.67	55.18
	Cross-LoRA	—	6.52	49.60	92.89	3.33	**23.33**	13.33	5.05	34.75	23.00	41.35
	TiTok (no seeds)	—	36.85	51.60	92.61	3.33	6.67	3.33	**40.91**	46.21	27.33	50.92
	Ours	GLM-5.3-Flash	**89.84**	**65.60**	81.83	**20.00**	6.67	10.00	15.15	28.84	20.33	57.29
	Ours	Qwen3.5-397B	82.49	53.80	**93.77**	6.67	3.33	**16.67**	36.36	**50.46**	**28.67**	**61.84**
Llama-8B	Base	—	12.13	14.00	79.10	0.00	0.00	0.00	29.80	9.43	9.33	24.80
	Native (labels)	—	42.76	21.80	**79.86**	0.00	0.00	**3.33**	27.27	5.55	7.67	31.53
	Cross-LoRA	—	3.87	22.20	79.67	0.00	**3.33**	0.00	29.29	14.42	13.00	26.63
	TiTok (no seeds)	—	23.20	22.60	79.56	0.00	0.00	0.00	30.30	11.46	10.33	29.43
	Ours	GLM-5.3-Flash	**52.46**	**24.00**	79.10	0.00	0.00	3.33	**31.82**	**18.11**	**14.00**	**37.53**
	Ours	Qwen3.5-397B	44.96	18.80	78.84	0.00	0.00	3.33	28.79	7.21	9.00	31.76

Our distillation and its ablations. Every probe in this paper is zero-prior. The 397B teacher first self-generates a probe pool from neutral seeds (2,286 probes; no benchmark data, task identity, or labels). Divergence scoring and coverage-driven evolution (§5.1, §5.2) turn that pool into the default elicited set (2,158 probes), which we call *ours*. Two ablations isolate the elicitation modules: *w/o div. & evo.* distills the self-generated pool directly, with neither divergence scoring nor evolution, and *w/o coverage* drops only the modality refill (2,071 probes). All variants share the same distillation protocol: the adapter-equipped teacher answers the probes and each student receives LoRA SFT on the resulting trajectories under an identical recipe.

6 Main results: train once, deploy many

Experimental setup. All students are evaluated under greedy decoding on full test sets: GSM8K and MATH-500 via Math-Verify, IFEval via official strict criteria, and IFBench via official prompt-level accuracy. We compare against unadapted base models (*Base*), supervised fine-tuning on original task labels (*Native*), and representative transfer baselines (Cross-LoRA, TiTok). The default teacher is Qwen3.5-397B, with GLM-5.3-Flash evaluated under an identical pipeline. All *ours* rows use the default elicited set (2,158 probes). *Mean* averages the five core benchmarks common to all rows (GSM8K, MATH-500, ARC-C, IFEval, and IFBench).

Scaling across student capacities. Across the Qwen family (Table 2), zero-prior distillation consistently outperforms native supervised training, maintaining a $+6.0$ to $+13.9$ point advantage on the five-benchmark mean. The gains concentrate in procedural reasoning and constraint adherence: on MATH-500 and IFEval, distilled students surpass native training at every capacity, elevating 9B MATH-500 to 68.20% ($+27.2$ over native). On out-of-distribution constraint following (IFBench), native training falls below base performance, whereas our distillation achieves robust improvements across all scales (e.g., 41.33% at 27B). Conversely, native training retains an advantage on GPQA-Diamond; divergence scoring over generic seeds naturally prioritizes procedural reasoning patterns over specialized factual recall.

Teacher flexibility. Replacing the default Qwen3.5-397B answering teacher with GLM-5.3-Flash under the identical pipeline confirms that elicitation is teacher-agnostic. Distilled students retain

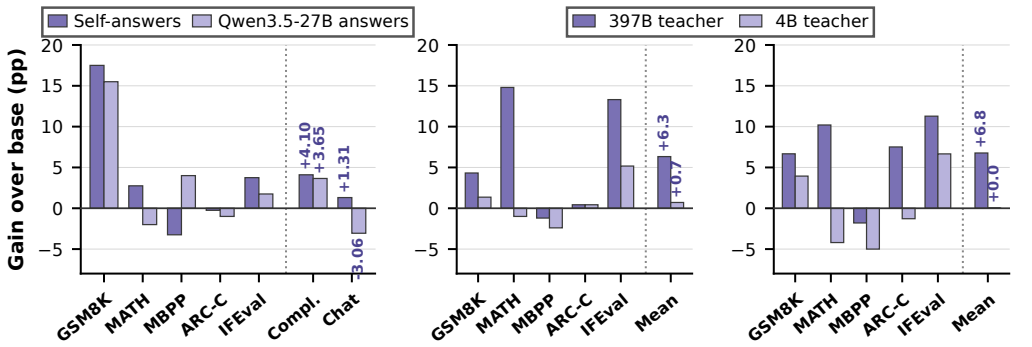
their gains over native training across all scales, with the 27B student reaching 40.00% on AIME 2024 and 36.67% on AIME 2025.

Comparison with transfer baselines. With the 397B teacher, our students beat both baselines on the mean at every student size (Table 2). Cross-LoRA's projected adapters break multi-step generation: GSM8K falls below base at every size (e.g., 28.43% vs. 88.25% at 27B). TiTok without seeds is the stronger baseline and scores well on GPQA-Diamond, but trails our students on GSM8K, MATH-500, and IFBench at every Qwen size, and the gap on the mean is largest at 4B (50.92 vs. 61.84).

Competition mathematics and stability. The AIME columns of Table 2 examine transfer to harder contest problems. Native training leaves combined AIME 2024+2025 accuracy at or below base levels. In contrast, zero-prior distillation lifts both 9B and 27B students to 20.0%, reaching 38.3% with the GLM teacher at 27B. At 4B and on Llama-3.1-8B, no arm moves more than two problems above base, indicating that elicitation amplifies reasoning capabilities the student already partially possesses rather than introducing them from scratch (30 problems/year, 3.3 points/problem; full ablation in Appendix E). Retraining across three seeds yields a standard deviation ≤ 0.3 on the five-benchmark mean (Appendix H), confirming that distillation gains exceed run-to-run noise.

Cross-family behavioral transfer. Text-mediated distillation bypasses parameter-basis incompatibility entirely (Table 2, bottom rows). On Llama-3.1-8B, zero-prior distillation delivers substantial gains in elementary reasoning (GSM8K $+32.8$ to $+40.3$ over base) and surpasses native fine-tuning on strict instruction following (18.11% vs. 5.55% on IFEval), demonstrating that the elicited behavioral distribution transfers across vocabularies and architectures.

Third-family transfer and teacher capacity. On Granite-3.3-8B-Instruct (Figure 3a), the zero-prior probe pool raises the completion macro by 4.10 points using the student's own greedy completions (and $+1.31$ chat macro), driven primarily by GSM8K ($+17.5$). That self-answers match or exceed answers from Qwen3.5-27B indicates that probe placement carries the primary signal. Conversely, teacher capacity is critical during elicitation (Figures 3b, 3c): replacing the 397B teacher with a 4B teacher collapses five-benchmark mean gains from > 6 points to ≤ 0.7 points, with MATH-500 dropping to base levels. The teacher backbone must possess sufficient capacity to express the adapter's capabilities upon probing.



(a) Third family: Granite-3.3-8B (b) Teacher scale: 27B student (c) Teacher scale: 9B student

Figure 3: **Third-family transfer and teacher scale**, as accuracy gains over each student's base model. (a) Granite-3.3-8B-Instruct trained on the frozen 1,200-probe zero-prior pool, answered either by the student itself or by Qwen3.5-27B; *Compl.* and *Chat* are the completion and chat macro averages on a disjoint held-out panel. (b, c) Qwen3.5 students distilled from the 397B or a 4B teacher on the self-generated probe pool (*w/o div. & evo.*); *Mean* is the five-benchmark mean. Full numbers are in Appendix F.

Table 3: Per-deployment cost (B200 GPU hours, wall-clock $\times$ 8 GPUs, measured from job logs). Teacher-side elicitation cost is one-time and amortizes over all targets. Native training additionally requires the 5,689 human-labeled examples to exist for every new deployment.

	Native LoRA (per target)	Ours
Human-labeled examples	5,689	**0**
Synthetic examples	0	2,158
Training tokens (3 ep.)	$\sim$2.4M	$\sim$0.78M
Steps	531	$\sim$200
Teacher one-time (amortized)	—	20.4
Target-side, 27B	57.1	**21.8**
Target-side, 9B	$\sim$31	**10.4**
Target-side, 4B	$\sim$28	**10.1**
Three-target total	$\approx$116	$\approx$**62.7**

7 IDENTIFICATION CONTROLS

Two explanations other than skill transfer could produce the gains in Table 2: the students might simply inherit the strength of a 397B teacher, or any fine-tuning data of similar size might help. We test both, and audit the elicited data for benchmark contamination.

Teacher strength vs. adapter skill. We regenerate answers to the evolved probes with the base 397B teacher, without the adapter, under identical decoding settings and length filters. The base teacher fails outright on 342 of the 2,071 probes in the *w/o coverage* set (16.5%). The probes selected by divergence therefore concentrate on behavior that the adapter adds to the teacher, rather than on questions the base teacher already handles.

Elicited data vs. generic fine-tuning. To confirm that the gains do not come from generic instruction tuning, we train the 9B student on a sample of Alpaca matched to the token count of our distillation data ($\sim$255K tokens) under an identical recipe. Generic instruction tuning lowers GSM8K accuracy from 78.62% (base) to 33.36%, so the volume of fine-tuning data does not explain the gains; what the students learn depends on the content of the elicited trajectories.

Contamination audit. We run a word-level 13-gram overlap audit, following the GPT-3 protocol (Brown et al., 2020), of all trajectory sets (prompts and completions) against every benchmark test set. The elicited curricula have zero 13-gram collisions with any benchmark. The only match, on MATH-500 in the self-generated pool, is a generic variable sequence ($x_1, x_2, x_3, y_1, y_2, y_3$) rather than a benchmark question. Surface n-gram overlap does not detect paraphrased problems (Yang et al., 2023; Golchin & Surdeanu, 2024), so this audit rules out verbatim leakage but not rephrased leakage.

8 COST ANALYSIS

Target-side distillation reduces training compute by 2.6–$3.0\times$ across all scales, requiring roughly one-third the training tokens and under half the gradient steps of native fine-tuning (Table 3). The one-time teacher elicitation cost (20.4 B200 GPU·h) amortizes rapidly as the number of downstream student targets grows, while bypassing the need to store, license, or collect human-labeled task data. Inference throughput benchmarks are reported in Appendix G.

9 CONCLUSION

Cross-model LoRA reuse does not require parameter compatibility. Transferring elicited behavioral distributions rather than weight deltas provides a scalable, annotation-free, and compute-efficient path to redeploy adapters across heterogeneous fleets.

Ethics statement

This work transfers task skills between models using synthetic data generated by the model owner's own adapter. The same elicitation primitive could, in principle, be used to extract capabilities from adapters one is licensed to query but not to copy. Our protocol, however, requires weight-level access to attach and detach the adapter, which typical API access does not provide. No human subjects or personal data are involved.

Reproducibility statement

All evaluations use public benchmarks with full test sets and greedy decoding; prompts, scoring scripts, elicited trajectory sets, probe pools, and student adapters will be released. The 397B teacher is proprietary; to support independent verification we include a teacher-scale ablation with a 4B teacher (§6).

The use of large language models

Large language models (LLMs) were used solely for text polishing, grammar checking, and minor language revisions. They were not used to generate experimental designs, scientific claims, analyses, or conclusions. All scientific content was authored and verified by the authors.

References

Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*, 2022.

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

Jan Betley, Xuchan Bao, Martín Soto, Anna Sztyber-Betley, James Chua, and Owain Evans. Tell me about yourself: Llms are aware of their learned behaviors. In *International Conference on Learning Representations*, volume 2025, pp. 21127–21179, 2025.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Michał Brzozowski, Zuzanna Dubanowska, Enrico Cassano, and Neo Christopher Chung. Reading the finetuning prior: Verbatim content recovery via contrastive decoding diffing. *arXiv preprint arXiv:2605.25902*, 2026.

Sheng Cao, Mingrui Wu, Karthik Prasad, Yuandong Tian, and Zechun Liu. Param

$$\backslash$$

delta $fordirectmixing : Post-trainlargelanguagemodelatzerocost.In$*International Conference on Learning Represen* $-48403, 2025.$

Rujikorn Charakorn, Edoardo Cetin, Yujin Tang, and Robert Tjarko Lange. Text-to-lora: Instant transformer adaption. *arXiv preprint arXiv:2506.06105*, 2025.

Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3513–3521. IEEE, 2019.

Lequn Chen, Zihao Ye, Yongji Wu, Danyang Zhuo, Luis Ceze, and Arvind Krishnamurthy. Punica: Multi-tenant lora serving. *Proceedings of Machine Learning and Systems*, 6:1–13, 2024.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

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

MAA Codeforces. American invitational mathematics examination-aime 2024, 2024.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

Farzad Farhadzadeh, Debasmit Das, Shubhankar Borse, and Fatih Porikli. Lora-x: Bridging foundation models with training-free cross-model adaptation. In *International Conference on Learning Representations*, volume 2025, pp. 9835–9858, 2025.

Shahriar Golchin and Mihai Surdeanu. Time travel in llms: Tracing data contamination in large language models. In *International Conference on Learning Representations*, volume 2024, pp. 43008–43029, 2024.

IBM Granite Team. Granite 3.0 language models. *URL: https://github. com/ibm-granite/granite-3.0-language-models*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*, 2023.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

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

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*, pp. 2790–2799. PMLR, 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. *arXiv preprint arXiv:2106.09685*, 2021.

Shih-Cheng Huang, Pin-Zu Li, Yu-Chi Hsu, Kuang-Ming Chen, Yu Tung Lin, Shih-Kai Hsiao, Richard Tsai, and Hung-yi Lee. Chat vector: A simple approach to equip llms with instruction following and model alignment in new languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10943–10959, 2024.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Chanjoo Jung and Jaehyung Kim. Titok: Transfer token-level knowledge via contrastive excess to transplant lora. In *International Conference on Learning Representations*, volume 2026, pp. 123084–123113, 2026.

Rana Muhammad Shahroz Khan, Pingzhi Li, Sukwon Yun, Zhenyu Wang, Shahriar Nirjon, Chau-Wai Wong Wong, and Tianlong Chen. Portllm: Personalizing evolving large language models with training-free and portable model patches. In *International Conference on Learning Representations*, volume 2025, pp. 36434–36451, 2025.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

Kalpesh Krishna, Gaurav Singh Tomar, Ankur P Parikh, Nicolas Papernot, and Mohit Iyyer. Thieves on sesame street! model extraction of bert-based apis. *arXiv preprint arXiv:1910.12366*, 2019.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Yanan Li, Fanxu Meng, Muhan Zhang, Shiai Zhu, Shangguang Wang, and Mengwei Xu. Lorasuite: Efficient lora adaptation across large language model upgrades. *Advances in Neural Information Processing Systems*, 38:170149–170176, 2026.

Zhiyuan Liang, Dongwen Tang, Yuhao Zhou, Xuanlei Zhao, Mingjia Shi, Wangbo Zhao, Zekai Li, Peihao Wang, Konstantin Schürholt, Damian Borth, et al. Drag-and-drop llms: Zero-shot prompt-to-weights. *Advances in Neural Information Processing Systems*, 38:153258–153283, 2026.

Pin-Jie Lin, Rishab Balasubramanian, Fengyuan Liu, Nikhil Kandpal, and Tu Vu. Efficient model development through fine-tuning transfer. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2617–2636, 2025.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A Smith. Tuning language models by proxy. *arXiv preprint arXiv:2401.08565*, 2024a.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*, 2024b.

Julian Minder, Clément Dumas, Caden Juang, Bilal Chughtai, and Neel Nanda. Overcoming sparsity artifacts in crosscoders to interpret chat-tuning. *Advances in Neural Information Processing Systems*, 38:106423–106474, 2026a.

Julian Minder, Clément Dumas, Stewart Slocum, Helena Casademunt, Cameron Holmes, Robert West, and Neel Nanda. Narrow finetuning leaves clearly readable traces in activation differences. In *International Conference on Learning Representations*, volume 2026, pp. 35340–35406, 2026b.

Eric Mitchell, Rafael Rafailov, Archit Sharma, Chelsea Finn, and Christopher D Manning. An emulator for fine-tuning large language models using small language models. *arXiv preprint arXiv:2310.12962*, 2023.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hanna Hajishirzi. Generalizing verifiable instruction following. *Advances in Neural Information Processing Systems*, 38, 2026.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Filippo Rinaldi, Aniello Panariello, Giacomo Salici, Angelo Porrello, and Simone Calderara. Transporting task vectors across different architectures without training. *arXiv preprint arXiv:2602.12952*, 2026.

Mikayel Samvelyan, Sharath C Raparthy, Andrei Lupu, Eric Hambro, Aram H Markosyan, Manish Bhatt, Yuning Mao, Minqi Jiang, Jack Parker-Holder, Jakob Foerster, et al. Rainbow teaming: Open-ended generation of diverse adversarial prompts. *Advances in Neural Information Processing Systems*, 37:69747–69786, 2024.

Ying Sheng, Shiyi Cao, Dacheng Li, Coleman Hooper, Nicholas Lee, Shuo Yang, Christopher Chou, Banghua Zhu, Lianmin Zheng, Kurt Keutzer, et al. S-lora: Serving thousands of concurrent lora adapters. *arXiv preprint arXiv:2311.03285*, 2023.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pp. 601–618, 2016.

Runqian Wang, Soumya Ghosh, David Cox, Diego Antognini, Aude Oliva, Rogerio Feris, and Leonid Karlinsky.

$$\backslash$$

textit {Trans-LoRA} : $towardsdata-freetransferableparameterefficientfinetuning.$*Advances in Neural Information* $61217 - -61237, 2024.$

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 13484–13508, 2023.

Feifan Xia, Mingyang Liao, Yuyang Fang, Defang Li, Yantong Xie, Weikang Li, Yang Li, Deguo Xia, and Jizhou Huang. Cross-lora: A data-free lora transfer framework across heterogeneous llms. *arXiv preprint arXiv:2508.05232*, 2025.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *International Conference on Learning Representations*, volume 2024, pp. 30745–30766, 2024.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. In *International Conference on Learning Representations*, volume 2025, pp. 76346–76382, 2025.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in neural information processing systems*, 36: 7093–7115, 2023.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E Gonzalez, and Ion Stoica. Rethinking benchmark and contamination for language models with rephrased samples. *arXiv preprint arXiv:2311.04850*, 2023.

Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8712–8721. IEEE, 2020.

Linyan 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 *Icml*, volume 2, pp. 21, 2024.

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.

A Cross-generation transplant and gauge diagnostics

Table 4 lists the full five-benchmark results behind Figure 1. Table 5 gives the GPQA comparison for the cross-generation transplant, Table 6 the Theseus-faithful α sweep, and Table 7 the Procrustes non-orthogonality diagnostics that explain why the gauge map cannot close the gap.

Additional cross-generation evaluations reinforce these findings: on the MBPP 17-item development set, adapter transplant achieves 88.24% accuracy, matching the native baseline without loss. In the reverse direction (Qwen3.6 $\rightarrow$ Qwen3.5), native training reaches 88.24% compared to 76.47% for direct transplant. On BigCodeBench-Hard, transplant incurs only a minor degradation of -0.7 percentage points, whereas the native baseline drops by -8.8 points. Under the strict 76-item GSM evaluation protocol, the transplanted adapter elevates accuracy from a base level of 26.32% to 86.84% ($\mathrm{TRR} = 0.958$, recovering 48 items with zero regressions), closely approaching the target-native ceiling of 89.47%.

Table 4: Cross-generation transplant (Qwen3.5-27B → Qwen3.6-27B), full five-benchmark evaluation, greedy. Superscripts give TRR against the Qwen3.6-native oracle; the MBPP oracle is below base (marked −). A separate strict 76-item GSM dev protocol gives $\mathrm{TRR} = 0.958$.

Arm	GSM8K	MATH-500	MBPP	ARC-C	IFEval	Mean
Qwen3.5-27B base	88.25	44.80	76.40	97.10	50.28	71.37
Qwen3.5-27B + native	93.03	60.80	73.60	97.78	59.33	76.91
Qwen3.6-27B base	52.92	15.20	74.60	97.27	36.04	55.21
Qwen3.6-27B + native (oracle)	93.78	**66.00**	70.40	97.44	57.12	**76.95**
Qwen3.6-27B + **zero-edit transplant**	**90.37**$^{0.92}$	63.40$^{0.95}$	72.40^{-}	**97.53**	**60.44**$^{1.16}$	76.83

Table 5: GPQA-Diamond (198 items) for the cross-generation transplant and the explicit gauge map (Q+s).

Arm	GPQA acc. (%)
Qwen3.5-27B base	34.3–34.9
Qwen3.5-27B + mix (native)	55.05
Qwen3.6-27B base	36.9–37.4
Qwen3.6-27B + zero-edit transplant	**48.48**
Qwen3.6-27B + gauge (Q+s, 24K tokens)	41.92
Qwen3.6-27B + gauge (Q+s, 200K tokens)	42.93

B Cross-family weight-space failure table

Controls: distilling the *no-LoRA* source's trajectories collapses to 1.0% (the skill is in the adapter); target self-distillation gives 10.0% ≈ base (no format effect). Reverse direction (Llama→Qwen) reaches $\mathrm{TRR} = 0.61$.

C Small-scale elicitation study and audits

Under a matched token budget, automatic probes (24.0–24.5%) match or surpass probes built from GSM questions (23.0%). A 2×2 factorial ablation attributes transfer gains additively to multi-step mutation (+6.5 points) and self-solvability filtering (+6.5 points); the dose-response sweep over the multi-step mutation fraction (Table 9) shows that multi-step complexity pressure is the dominant factor. A zero-prior audit removes all manually specified mathematics priors (<<>> regex patterns, arithmetic character sets, few-shot exemplars, and domain tag filters); the pipeline autonomously re-derives the motif specification from base-vs-adapted trajectory contrast, achieving 27.0% with 289 probes compared to 27.5% under manually specified constraints. Regarding domain limits: MBPP distillation at this compact scale is bounded by the oracle performance ceiling (7.2%), and general Alpaca conversational transfer exhibits limited gains ($\mathrm{TRR}_{\mathrm{rouge}} = 0.16$). Perplexity evaluations indicate modest shifts in general language modeling: WikiText perplexity moves from 8.23 (base) to 9.58 under full distillation.

D Elicited trajectory statistics

E Competition mathematics without divergence scoring and evolution

Table 11 compares our default curriculum with the *w/o div. & evo.* ablation, which distills the teacher's self-generated probe pool directly. On the 9B student, divergence scoring and evolution account for the AIME gain (20.0% combined, against 8.3% for the ablation); on the 27B student the two are within one problem of each other, and at 4B neither rises above base. The last block attaches the distilled Qwen3.5-27B adapters to Qwen3.6-27B with zero edits. There the result reverses: the

Table 6: Theseus-faithful α sweep ($\theta + \alpha\tau$) and split-half protocol (even items select α, odd items report). Zero-edit dominates the gauge formula at every α.

Arm	α=0.25	0.5	0.75	1.0	split-half
zero-edit	—	41.4	44.4	**48.0**	**46.5**
gauge (Q+s)	36.9	37.9	41.9	44.4	42.4

Table 7: Procrustes non-orthogonality $\|Q - I\|/\sqrt{2d}$ by interface dimension: near-identity in small subspaces, non-orthogonal drift in FFN space; $8\times$ more tokens improves the fit but adds only $\sim$1pp GPQA. Scale $s = 0.98 \pm 0.03$ throughout (no gain drift).

Dimension	24K tokens	200K tokens
48 (interface)	0.072	0.065
5120 (residual stream)	0.319	0.178
17408 (FFN)	0.859	0.748

ablation adapter reaches 13.3% combined, above the native arm's 8.3%, while the default adapter drops to 1.7%.

F THIRD-FAMILY TRANSFER AND TEACHER SCALE: FULL RESULTS

Tables 12 and 13 give the numbers behind Figure 3. The teacher-scale gains in the figure are measured against the base rows of Table 13.

G THROUGHPUT

H MULTI-SEED REPLICAS AND NOISE FLOOR

We retrain three distillation arms (9B w/o coverage, 27B w/o coverage, and 27B w/o div. & evo.) under three training seeds each and report the mean and standard deviation of the five-benchmark mean in Table 15. To separate seed variance from system nondeterminism, we also repeat runs with an identical seed; these duplicates differ by at most 0.5 points per benchmark. Across seeds, the standard deviation of the five-benchmark mean is 0.2–0.3 points, an order of magnitude smaller than the 6–14 point gaps between our distillation and native training in Table 2.

Table 8: Qwen2.5-3B → Llama-3.2-3B, GSM8K. All weight-space variants fail; the capacity control shows the bottleneck is supervision, not LoRA expressiveness. Small-scale study.

Method	acc. (%)	TRR
joint functional Qr_s (wikitext probes)	4.5	-0.13
joint functional Qr_s (GSM probes)	0.0	-0.27
behavior delta (scalar / multi-candidate)	0–7	≤ -0.05
decision/dense field trust-region	0.5–5	< 0
learned compiler G_ϕ (24M params, loss 0.002)	10.5	0.06
capacity control (target-native hidden states)	27.0	0.59
distillation, 384 unlabeled questions	32.6	0.79
distillation, 7473 unlabeled questions	50.2	**1.34**
distillation, 1200 zero-prior probes	35.03	**0.845**

Table 9: Dose-response of small-scale student accuracy over the multi-step mutation fraction in the elicitation operators (GSM dev protocol).

Multi-step fraction	0	0.25	0.5	0.75	1.0
Student accuracy (%)	18.5	24.0	**34.5**	26.5	34.0

Table 10: Modality histograms of elicited trajectory sets. Default ours uses two-level stratified selection with scarcest-modality refill; the w/o-coverage ablation drops that refill and remains math-concentrated.

Version	Traj.	Skeletons	code	math	qa	instr.	choice
w/o coverage	2071+100	304	1	∼1100	253	76	0
ours (default)	2158+100	215	355	759	364	275	112

Table 11: AIME 2024 / 2025 accuracy (%) of our default curriculum and the *w/o div. & evo.* ablation (30 problems per year; completion prompt; integer match in $\{0, \ldots, 999\}$). Base, native, and GLM rows repeat Table 2. The Qwen3.6-27B block uses the Qwen3.5-27B distilled adapters attached with zero edits.

Student	Arm	AIME24	AIME25
27B	base	10.0	3.3
	native (labels)	6.7	6.7
	ours	23.3	16.7
	ours (GLM-5.3-Flash)	**40.0**	**36.7**
	w/o div. & evo.	23.3	13.3
9B	base	3.3	3.3
	native (labels)	0.0	0.0
	ours	23.3	**16.7**
	ours (GLM-5.3-Flash)	**26.7**	13.3
	w/o div. & evo.	10.0	6.7
4B	base	6.7	**13.3**
	native (labels)	6.7	3.3
	ours	6.7	3.3
	ours (GLM-5.3-Flash)	**20.0**	6.7
	w/o div. & evo.	6.7	3.3
Qwen3.6-27B	base	0.0	0.0
	native (labels)	**10.0**	6.7
	ours	3.3	0.0
	w/o div. & evo.	6.7	**20.0**

Table 12: Third-family transfer (Granite-3.3-8B-Instruct (Granite Team, 2024)), disjoint held-out panel (offset 100, n=400 per task). *Compl. macro* is the five-task completion mean and *Chat macro* the four-task chat mean. *Self-answers*: the student's own greedy answers on the frozen 1,200-probe zero-prior pool; *Qwen3.5-27B answers*: Qwen3.5-27B's answers to the same probes. Both arms use three-epoch full-parameter response-only SFT with an identical recipe and no task identity, seeds, labels, or benchmark examples.

Arm	GSM8K	MATH-500	MBPP	ARC-C	IFEval	Compl. macro	Chat macro
base	20.50	35.25	5.50	**79.25**	56.25	39.35	62.44
Qwen3.5-27B answers	36.00	33.25	**9.50**	78.25	58.00	43.00	59.38
self-answers	**38.00**	**38.00**	2.25	79.00	**60.00**	**43.45**	**63.75**

Table 13: Teacher-scale ablation: the 397B teacher replaced by a 4B teacher under the identical recipe, both distilling the self-generated probe pool (*w/o div. & evo.*); five-benchmark protocol, *Mean* includes MBPP.

Student	Teacher	GSM8K	MATH-500	MBPP	ARC-C	IFEval	Mean
27B	none (base)	88.25	44.8	**76.4**	97.10	50.28	71.37
	397B	**92.57**	**59.6**	75.2	**97.53**	**63.59**	**77.70**
	4B	89.61	43.8	74.0	**97.53**	55.45	72.08
9B	none (base)	78.62	29.6	**62.4**	80.63	42.88	58.83
	397B	**85.29**	**39.8**	60.6	**88.14**	**54.16**	**65.60**
	4B	82.56	25.4	57.4	79.35	49.54	58.85

Table 14: Greedy generation throughput (B200, tok/s; vLLM serving (Kwon et al., 2023)) from the main-table evaluation; distilled students generate longer chains-of-thought on GSM8K, hence lower tok/s than base.

Student	Weights (GB)	GSM8K tok/s (base/native/distill)	MATH tok/s (base/native/distill)
27B	51.7	2245 / 1771 / 1780	7149 / 6630 / 6989
9B	18.0	8307 / 5358 / 6141	24059 / 17882 / 21671
4B	8.7	7203 / 5633 / 6962	20624 / 16446 / 17141

Table 15: Multi-seed replicas of distillation ablations (three training seeds; five-benchmark protocol, mean includes MBPP; mean $\pm$ std).

Student	Arm	5-bench mean
9B	w/o coverage	63.5 ± 0.2
27B	w/o coverage	77.5 ± 0.3
27B	w/o div. & evo.	78.0 ± 0.3