ATTUNE: STATE-ADAPTIVE SUPERVISION ALLOCATION FOR ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation trains compact language models with teacher feedback on their own responses. Deciding which feedback to emphasize remains challenging: teacher–student disagreement signals a prediction gap, while the benefit of correcting it depends on the student's capacity and current training state. We introduce a controlled paired-intervention protocol using supervised fine-tuning on compositional arithmetic to measure the returns of fixed supervision under matched token exposure. The relative gains vary across student capacities and training states, supporting a learner-conditioned view of supervision utility. We operationalize this view through Attune, a coupled token–group allocation framework for on-policy distillation. Applying the same token calibration to candidate and reference losses couples the learning signal to the gradient comparisons used for allocation. Across eight benchmarks and two model families, Attune achieves higher aggregate scores than the compared distillation methods, and these gains persist across three training seeds. Independent-probe interventions further show that both components reduce distillation loss relative to uniform and shuffled weighting. Attune requires no difficulty labels and adds no inference-time computation.

1 INTRODUCTION

Knowledge distillation (KD) transfers the capabilities of large language models (LLMs) to smaller students with lower inference costs (Hinton et al., 2015). On-policy distillation obtains teacher feedback on student-generated responses, allowing supervision to follow the contexts that the learner encounters (Agarwal et al., 2024; Gu et al., 2024). Yet obtaining feedback on the student's own trajectories leaves open how strongly each signal should influence training. Effective transfer therefore calls for allocating learning effort to supervision that benefits the student at its current training state (cf. Wang et al., 2026b).

Curriculum and self-paced learning adjust training emphasis using example difficulty and the learner's current state (Bengio et al., 2009; Kumar et al., 2010). Student-aware distillation approaches adapt supervision through difficulty-aware data selection (He et al., 2025), competence-based curricula (Huang et al., 2026; Xu et al., 2026a), and token- or trajectory-level selection and reweighting (Xu et al., 2026b; Wang et al., 2026c; Li et al., 2026). These approaches use properties of the data and the student's predictions to guide learning. A central challenge is to connect such signals to the benefit of an update: a difficult example or a large teacher–student disagreement may call for attention, but the return from emphasizing it depends on the receiving student. As learning progresses, the student can also gain more from supervision that was less useful earlier. An allocation rule that misses such changes can continue emphasizing supervision with lower utility while underusing newly beneficial signals. This raises the question of how supervision utility varies across students and training states, and how allocation can respond to those changes.

We introduce a controlled paired-intervention protocol to measure the returns of fixed supervision across student capacities and training states (Section 2). Using supervised fine-tuning on compositional arithmetic, we apply fixed basic and difficult supervision blocks to four student capacities at multiple training states under matched token exposure. At the endpoint, the larger students gain more from difficult than from basic supervision, whereas the smallest favors basic supervision on

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.

average. The relative gains also change within each student as training proceeds. Fixed supervision thus yields changing utility, supporting a learner-conditioned view of its value.

We introduce Attune, a coupled token–group allocation framework for on-policy distillation. Token calibration shapes both how a response group contributes to learning and how its update is scored for allocation. Token calibration can change a group's gradient direction, so scoring uncalibrated gradients may assess a different update from the one used for learning. Attune therefore compares calibrated candidate gradients with a similarly calibrated reference gradient, refreshed on current student responses, to assign bounded group weights. Gradient compatibility provides a local utility surrogate, drawing on gradient-based reweighting and data selection (Ren et al., 2018; Xia et al., 2024; Yang et al., 2026). Attune retains every response token, requires no difficulty labels, and adds no inference-time computation.

Across eight benchmarks spanning five capability groups, Attune achieves the highest aggregate scores among the compared distillation methods in both Qwen2.5 and Llama-3.2. Three-seed paired comparisons show TaskAvg gains of 2.04 and 2.08 points over vanilla GKD, respectively, and gains over the selected method baselines. Independent-probe interventions show that token calibration and group allocation yield larger one-step reductions in distillation loss than uniform and shuffled controls. Component ablations further support applying token calibration to the gradients used for allocation.

Our contributions are threefold:

- We introduce a controlled paired-intervention protocol that varies dependency depth while matching operation count, target-trace length, and token exposure, revealing capacity- and state-dependent returns from fixed supervision in compositional arithmetic.
- We propose Attune, a coupled token–group allocation framework in which token calibration shapes both the learning signal and the gradient comparisons used to allocate on-policy supervision.
- We demonstrate aggregate performance gains across two model families and eight benchmarks, supported by three-seed paired comparisons and local interventions that test the allocation mechanism.

2 A Controlled Diagnosis of State-Dependent Supervision Utility

We introduce a paired-intervention protocol to measure how the returns of fixed supervision vary with student capacity and training state. The diagnosis uses full-parameter SFT on fixed target traces and is independent of Attune's allocation mechanism.

2.1 Controlled Arithmetic and Paired Interventions

We use a compositional-arithmetic task in which every example contains 15 operations and a fixed 15-line target trace. Its five difficulty levels vary the length of the critical dependency path while holding operation counts and trace length fixed. Levels 1–3 form the *anchor* slice and Levels 4–5 form the *hard* slice. This construction separates dependency depth from nominal computation and supervised-token exposure.

We evaluate official Qwen2.5-Instruct students at 0.5B, 1.5B, 3B, and 7B parameters under three paired training seeds on a fixed held-out probe of 500 examples, balanced across the five difficulty levels. The primary metric is final-answer accuracy, measured by numeric equality modulo 23 under greedy generation; invalid outputs count as incorrect. We additionally report intermediate-step accuracy as a process diagnostic.

We use a single-pass counterfactual design to separate training state from repeated exposure. Each parent run traverses the anchor-only stream once and saves states at 0%, 25%, 50%, 75%, and 100% of its unique-example budget. At each state θ_t, two independent copies receive either a 512-example anchor block or a response-token-matched 512-example hard block. Response-token matching controls the amount of supervision when comparing the anchor and hard branches. For each seed, we

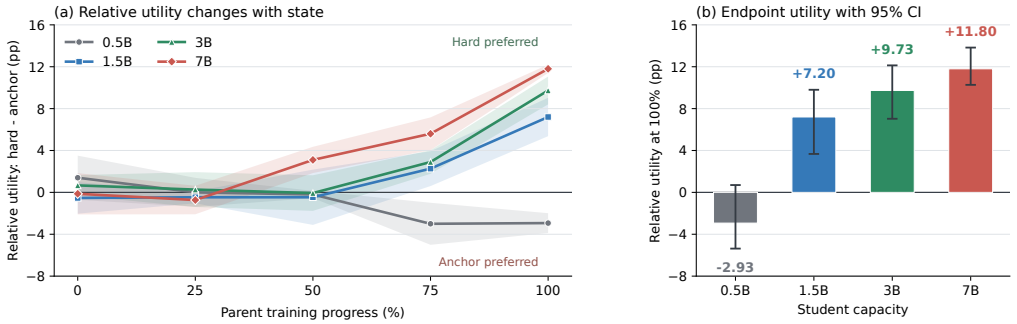


Figure 1: **Fixed supervision, changing utility.** (a) Hard-minus-anchor utility $R(t)$ over a single-pass parent stream; lines and ribbons show the mean and sample SD over three paired seeds. (b) Endpoint bars show the means, with whiskers indicating paired bootstrap 95% confidence intervals. Positive values favor hard supervision.

pre-register one pair (A_s, H_s), used with the same within-block order at all five states and all four capacities. Each independent branch processes its assigned block once. The parent stream, intervention pairs, and probe are mutually disjoint. Appendix A.2 provides the data partitions, arithmetic construction, training budgets, and token-matching measurements.

For seed s and parent state t, let $\theta^{(s)}_{t,A}$ and $\theta^{(s)}_{t,H}$ denote the checkpoints after the anchor and hard interventions, respectively. With M denoting held-out final-answer accuracy, we define relative utility as

$$R^{(s)}(t) = M\left(\theta^{(s)}_{t,H}\right) - M\left(\theta^{(s)}_{t,A}\right). \tag{1}$$

Positive values favor hard supervision; negative values favor anchor supervision. Absolute marginal utility is each branch's accuracy gain over its parent checkpoint. Reported results average the paired quantities across seeds.

2.2 Difficult Supervision Has Capacity-Dependent Returns

At the endpoint, hard supervision outperforms anchor supervision for all three larger students, with paired bootstrap 95% intervals entirely above zero (Figure 1b; Appendix Table 6). The 0.5B student instead favors anchor in all three seeds, although its interval includes zero. The return from the same matched supervision thus depends on the receiving student's capacity.

2.3 The Preferred Supervision Changes with Training State

Figure 1a shows how this capacity dependence emerges during training. The 1.5B and 3B students develop a preference for hard supervision after 50% progress, while 7B does so earlier. In contrast, 0.5B becomes anchor-preferred late in training.

Both interventions yield positive mean gains over their parents late in training, including hard supervision for 0.5B (Figure 2; Appendix A.2).

With supervision content and exposure fixed, these changing relative gains show that utility depends on the interaction between supervision and the current learner.

2.4 Implications for Adaptive Supervision

For the 0.5B student, improved intermediate-step accuracy under hard supervision does not ensure better final-answer accuracy (Appendix A.2).

The fixed-block interventions support a learner-conditioned view of supervision utility: its value depends on the receiving student's capacity and training state. Complementary experiments across

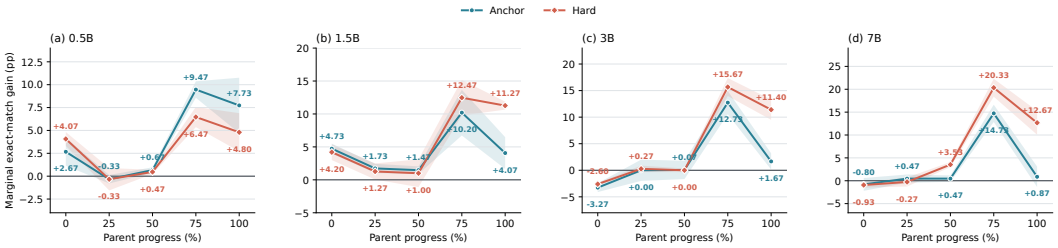


Figure 2: **Absolute marginal utility of fixed anchor and hard supervision.** Labels show mean exact-match gains over the parent checkpoint in percentage points; ribbons show ± 1 sample SD over three seeds. Vertical scales differ across panels.

procedural arithmetic, natural mathematics, and deductive reasoning show domain-dependent returns (Appendices A.3–A.4). Under the reported training and generation budgets, mixed supervision lowers teacher-forced NLL at every capacity yet reduces mean aggregate free-generation accuracy for the two smaller students relative to anchor-only training. These results distinguish supervision difficulty, teacher-forced fit, and realized capability gains.

Attune turns calibrated gradient compatibility with a reference KD loss into an online allocation rule (Section 3). Sections 4.4 and 4 evaluate its local probe-loss effects and downstream performance, respectively.

3 Attune: State-Adaptive Supervision Allocation

3.1 Overview

Attune couples token calibration and group allocation, so that token weights shape both the final update loss and the gradients used to score candidate groups. It compares calibrated candidate and reference gradients as a local utility surrogate, avoiding separate trial updates for each group. Compatibility is measured in selected parameter subspaces, while the weighted loss updates all student parameters.

3.2 Token-Calibrated On-Policy Updates

Let π_θ be the student and π_{T} a fixed teacher. Given a prompt x_i, the current student samples a response $y_i \sim \pi_\theta(\cdot \mid x_i)$. At each response position u, the teacher and student produce next-token distributions $p^{\mathrm{T}}_{i,u}$ and $p^{\mathrm{S}}_{i,u}$ on the same student-generated context. We use their generalized Jensen–Shannon divergence with coefficient 0.5 as the base token loss $\ell^{\mathrm{KD}}_{i,u}$ (Agarwal et al., 2024). Gradients do not pass through discrete generation.

At step t, the global training batch is partitioned into B groups. Let $\mathcal{M}_{t,b}$ be the valid response positions in group $G_{t,b}$ and $N_{t,b} = |\mathcal{M}_{t,b}|$. For each realized student token, define the one-sided log-probability gap $d_{i,u} = [\log(p^{\mathrm{T}}_{i,u}(y_{i,u})/p^{\mathrm{S}}_{i,u}(y_{i,u}))]_+$ and the raw pace weight $r_{i,u} = \max(\exp(-\gamma d_{i,u}), \epsilon)$. Let $\mu_{t,b}$ denote the mean raw weight over $\mathcal{M}_{t,b}$. We normalize within each group and shrink toward uniform weighting:

$$
\begin{aligned}
a_{i,u} &= (1 - \lambda_{\mathrm{tok}}) + \lambda_{\mathrm{tok}} \frac{r_{i,u}}{\mu_{t,b}}, \\
\mathcal{L}^{\mathrm{cal}}_{t,b} &= \frac{1}{N_{t,b}} \sum_{(i,u) \in \mathcal{M}_{t,b}} \mathrm{sg}(a_{i,u}) \ell^{\mathrm{KD}}_{i,u}.
\end{aligned}
\tag{2}
$$

Here sg denotes stop-gradient. Larger positive gaps receive smaller raw weights; nonpositive log gaps receive $r_{i,u} = 1$. With $\gamma = 0.25$ and $\lambda_{\mathrm{tok}} = 0.25$, the normalized weights have mean one and satisfy $a_{i,u} > 0.75$, keeping every response position active. The same rule independently calibrates the controller loss below; implementation details appear in Appendix A.1.

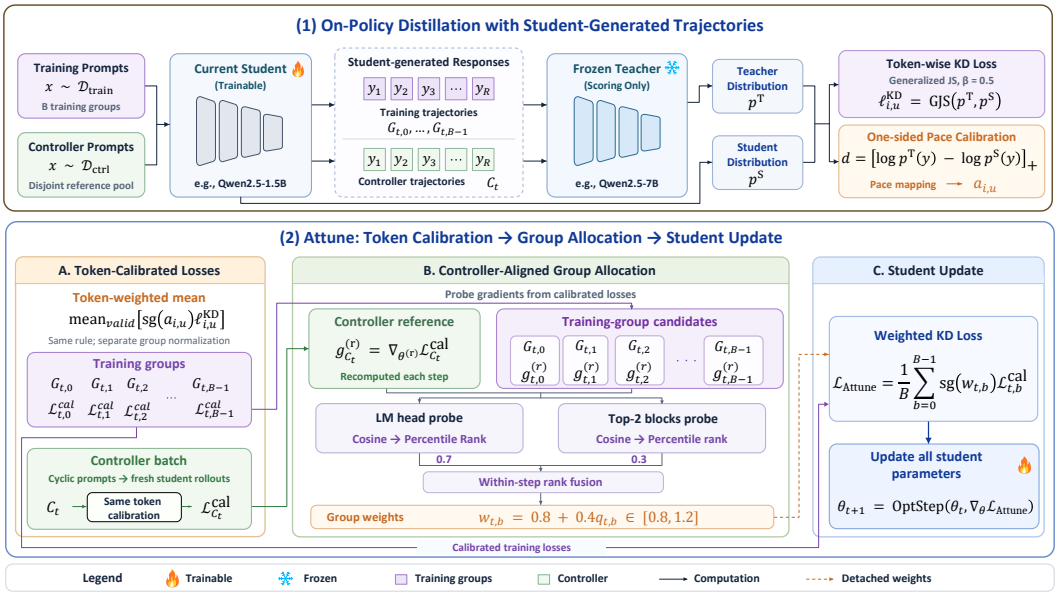


Figure 3: Overview of Attune. Token calibration conditions the gradients used on both sides of the comparison; controller alignment then reallocates optimization effort across candidate training groups.

3.3 Controller-Aligned Group Allocation

Controller direction. A controller split, disjoint from ordinary training and development data, provides a common reference from the instruction mixture. At each step, the student generates fresh responses for a controller minibatch C_t using the training decoding policy, and the teacher scores the same trajectories. Applying Equation 2 independently within C_t defines a calibrated controller loss $\mathcal{L}^{\mathrm{cal}}_{C_t}$, which guides training solely through the allocation weights. Pool construction and sampling are specified in Appendix A.1.

Directional compatibility. Let $g_{C_t} = \nabla_\theta \mathcal{L}^{\mathrm{cal}}_{C_t}$ and $g_{t,b} = \nabla_\theta \mathcal{L}^{\mathrm{cal}}_{t,b}$. Holding the sampled trajectories and detached gates fixed, a small update $-\eta g_{t,b}$ has a first-order controller-loss decrease of $\eta\langle g_{C_t}, g_{t,b}\rangle$. Cosine normalization removes gradient scale, and restriction to the language-model head and final two Transformer blocks further approximates the full-space comparison. The resulting score measures directional compatibility; predicting an AdamW step additionally depends on optimizer state and the full batch gradient. For probe space r, the score is

$$s^{(r)}_{t,b} = \frac{\langle g^{(r)}_{C_t}, g^{(r)}_{t,b}\rangle}{\|g^{(r)}_{C_t}\|_2 \, \|g^{(r)}_{t,b}\|_2}. \tag{3}$$

Bounded relative allocation. At each step, we convert the cosine scores in each probe space to percentile ranks $q^{(r)}_{t,b} \in [0,1]$, using average ranks for ties, and compute

$$\begin{aligned} q_{t,b} &= \rho q^{(\mathrm{head})}_{t,b} + (1-\rho) q^{(\mathrm{top})}_{t,b}, \\ w_{t,b} &= 1 + \lambda_{\mathrm{grp}}(2q_{t,b} - 1). \end{aligned} \tag{4}$$

We use $\rho = 0.7$ and $\lambda_{\mathrm{grp}} = 0.2$, yielding group weights with mean one in $[0.8, 1.2]$. These weights keep every group active and adjust its contribution according to relative compatibility within the batch.

3.4 JOINT OBJECTIVE

The final loss combines the two allocation levels:

$$\mathcal{L}_{\text{Attune}} = \frac{1}{B}\sum_{b=1}^{B}\frac{\text{sg}(w_{t,b})}{N_{t,b}}\sum_{(i,u)\in\mathcal{M}_{t,b}}\text{sg}(a_{i,u})\ell_{i,u}^{\text{KD}}. \tag{5}$$

Setting $\lambda_{\text{tok}} = \lambda_{\text{grp}} = 0$ recovers distributed vanilla GKD exactly. One backward pass through Equation 5 updates all student parameters; controller and probe computations add no inference overhead. In the full Qwen2.5 7B-to-1.5B runs, total model FLOPs are $1.242\times$ those of vanilla GKD. The aggregate gains over GKD persist in training-step budget comparisons guided by this FLOP ratio (Appendix A.11). Algorithm 1 in Appendix A.1 gives the complete procedure.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

The full baseline comparison (Table 1) and component ablations (Table 3) use training seed 10. We separately report three-seed paired comparisons against selected baselines in Table 2, with training runs as the statistical units (Appendix A.6).

Models, data, and training. Our main results cover two within-family teacher–student pairs: Qwen2.5-7B-Instruct → Qwen2.5-1.5B-Instruct (Qwen Team, 2024) and Llama-3.2-3B-Instruct → Llama-3.2-1B-Instruct (Meta, 2024). For the main comparisons, we use a mixture of 100,000 English prompts from public instruction-following and mathematics datasets, partitioned into 96,000 training, 2,000 controller, and 2,000 development prompts with disjoint splits. Source composition and filtering are described in Appendix A.6. For vanilla GKD and Attune, the frozen teacher scores student-generated trajectories. The primary Qwen configuration uses a global prompt batch of 64, 1,500 optimizer steps, and a maximum generated response length of 8,192 tokens. Vanilla GKD and Attune use generalized Jensen–Shannon divergence with $\beta = 0.5$ (Agarwal et al., 2024). Experiments in both model families run on eight NVIDIA RTX6000 Blackwell GPUs.

Method configuration. Here Attune always denotes the complete token-plus-group method. In the Qwen configuration, the token gate softly rescales realized-token losses, while the group controller compares each of eight training-group gradients with a held-out controller gradient in the LM head and final two Transformer blocks. The global controller minibatch contains 32 reference prompts per step. Fused percentile ranks produce nonzero weights in $[0.8, 1.2]$. Appendix A.6 lists the optimizer, rollout, precision, and controller settings.

Evaluation. We evaluate on eight benchmarks: IFEval (Zhou et al., 2023), MMLU-Pro (Wang et al., 2024), TruthfulQA-MC2 (Lin et al., 2022), DROP (Dua et al., 2019), MuSR (Sprague et al., 2024), BBH (Suzgun et al., 2023), GSM8K (Cobbe et al., 2021), and MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024). All scores use the task-specific metric on a 0–100 scale. *TaskAvg* averages the eight tasks; *CatAvg* gives equal weight to five capability groups. The full aggregation formula is given in Appendix A.6, Equation 6. Both model families use the same frozen eight-task evaluation protocol and evaluation seed 42. Dataset revisions, prompting, decoding, and answer extraction for both model families are specified in Appendix A.6.

4.2 MAIN RESULTS AND BASELINES

We compare Attune with the official student, teacher-response SFT (Kim & Rush, 2016), and vanilla GKD in Table 1. The baselines also include objective and trajectory alternatives—FKL, RKL (Gu et al., 2024), Bridge-Garden Hybrid KD (Wang et al., 2026a), DistiLLM-2 (Ko et al., 2025), and SKD (Xu et al., 2025a)—and the student-aware allocation methods TIP (Xu et al., 2026b), TA-OPD (Wang et al., 2026c), and FiRe-OPD (Li et al., 2026). Appendix A.7 describes the method-specific training settings and response construction.

Table 1: **Results on eight benchmarks across two model families.** Evaluation seed 42; higher is better. Bold/underlining mark the best/second-best within each family. Aggregates use displayed scores before rounding; CatAvg follows Equation 6. Three-seed summaries appear in Table 2.

Method	IFEval	MMLU-Pro	TQA-MC2	DROP	MuSR	BBH	GSM8K	MATH-500	**TaskAvg**	**CatAvg**
(a) Qwen2.5: 7B → 1.5B										
Official student	40.67	31.57	46.58	7.06	34.51	43.97	65.28	47.80	39.68	36.52
Teacher-response SFT	42.70	30.49	46.88	5.94	35.05	41.08	62.40	47.40	38.99	36.06
Vanilla GKD	53.05	31.98	50.12	7.14	36.24	44.92	65.88	48.60	42.24	39.81
On-policy FKL	52.68	31.03	49.32	6.63	35.71	44.91	66.79	48.40	41.93	39.48
On-policy RKL	55.27	31.50	49.30	6.45	37.10	44.08	65.43	49.40	42.32	40.03
Bridge-Garden Hybrid KD	53.79	31.28	49.52	6.79	36.77	43.86	67.63	49.00	42.33	39.92
DistiLLM-2	53.23	32.87	48.14	7.20	36.64	45.34	66.34	50.40	42.52	40.06
SKD (top-25)	51.20	32.96	49.74	7.19	36.64	43.20	65.81	49.80	42.07	39.49
TIP (Soft-OR, 50%)	54.71	30.00	49.69	7.30	36.64	45.75	63.99	50.20	42.29	40.03
TA-OPD (K=16, 10%)	55.08	32.41	50.20	7.21	36.89	44.54	65.88	49.20	42.68	40.37
FiRe-OPD	52.68	32.24	49.30	7.23	36.34	44.89	66.34	50.00	42.38	39.89
Attune	**56.38**	**33.74**	**50.60**	**7.32**	**37.83**	**46.71**	**70.43**	**51.40**	**44.30**	**41.81**
(b) Llama-3.2: 3B → 1B										
Official student	44.92	16.97	40.51	6.28	28.75	36.04	45.49	19.20	29.77	28.94
Teacher-response SFT	44.55	16.61	40.88	5.94	28.05	37.08	44.35	20.60	29.76	28.86
Vanilla GKD	48.24	16.64	41.38	6.45	29.28	39.04	44.28	25.80	31.39	30.58
On-policy FKL	46.21	17.33	41.41	6.28	29.48	40.15	42.46	24.40	30.97	30.02
On-policy RKL	47.50	17.51	41.21	6.19	29.01	41.13	45.49	24.20	31.53	30.59
Bridge-Garden Hybrid KD	47.50	16.87	41.12	5.64	29.69	40.84	44.35	25.00	31.38	30.42
DistiLLM-2	48.43	17.16	40.81	6.17	**30.42**	40.78	46.17	25.80	31.97	31.03
SKD (top-25)	46.77	17.25	41.10	6.28	29.29	41.36	47.69	24.20	31.74	30.70
TIP (Soft-OR, 50%)	46.21	18.12	41.49	6.35	29.56	39.92	48.07	25.40	31.89	30.77
TA-OPD (K=16, 10%)	46.40	17.06	41.70	6.51	29.82	40.32	47.16	25.40	31.80	30.73
FiRe-OPD	48.24	18.76	40.78	6.45	29.42	39.13	48.29	24.00	31.88	30.98
Attune	**50.46**	**19.97**	**43.17**	**6.61**	30.03	**41.39**	**49.20**	**26.20**	**33.38**	**32.41**

Table 2: **Aggregate results over three training seeds.** Attune scores are mean ± SD; Δ denotes the mean paired gain. Selected baselines are TA-OPD for Qwen and DistiLLM-2 for Llama. CIs are unadjusted paired-t intervals; full comparisons appear in Table 15.

Family	Metric	Attune	Δ_{GKD}	Δ_{sel}	95% CI (Δ_{sel})
Qwen	TaskAvg	44.17 ± 0.14	$+2.04$	$+1.54$	$[1.25, 1.84]$
	CatAvg	41.70 ± 0.14	$+1.97$	$+1.35$	$[1.07, 1.64]$
Llama	TaskAvg	33.49 ± 0.11	$+2.08$	$+1.54$	$[1.26, 1.82]$
	CatAvg	32.55 ± 0.13	$+1.92$	$+1.49$	$[1.24, 1.74]$

Performance across model families. In the seed-10 comparison, Attune achieves the highest TaskAvg and CatAvg in both model families, reaching 44.30/41.81 for Qwen and 33.38/32.41 for Llama. It leads all eight tasks for Qwen and seven of eight for Llama; DistiLLM-2 leads on Llama MuSR. TA-OPD and DistiLLM-2 are the strongest non-Attune aggregate baselines for Qwen and Llama, respectively, and serve as the selected method baselines in the repeated-run comparisons below.

Variation across training seeds. Across three training seeds, Attune improves both aggregates over vanilla GKD and the selected baseline in each family (Table 2); all eight aggregate comparisons remain significant after Holm correction. Task-level gains vary, with no clear advantage over

DistiLLM-2 on Llama MuSR; complete results are in Table 15 in the appendix. Additional results for the smaller Qwen2.5-0.5B student appear in Appendix A.10.

4.3 Coupling Token Calibration and Group Allocation

We test whether token calibration should shape the gradients used for allocation as well as the final update loss, alongside comparisons of allocation levels and gradient spaces (Table 3).

Table 3: **Component comparisons for Qwen distillation on eight benchmarks.** Token rule refers to token weighting in the final student-update loss.

Variant	Token rule	Group rule	IFEval	MMLU-Pro	TQA-MC2	DROP	MuSR	BBH	GSM8K	MATH-500	TaskAvg	CatAvg
Vanilla GKD	Uniform	Uniform	53.05	31.98	50.12	7.14	36.24	44.92	65.88	48.60	42.24	39.81
Token gate only	Gap gate	Uniform	54.16	32.50	49.86	7.22	36.42	45.92	66.34	49.00	42.68	40.28
Group alignment only	Uniform	Head+Top (online)	55.64	32.93	50.94	7.28	37.43	45.69	67.70	50.00	43.45	41.05
Full Attune	**Gap gate**	**Head+Top (online)**	**56.38**	**33.74**	**50.60**	**7.32**	**37.83**	**46.71**	**70.43**	**51.40**	**44.30**	**41.81**
Uncalibrated alignment	Gap gate	Head+Top (uncalibrated)	55.08	32.87	50.25	7.21	36.49	45.52	66.64	50.00	43.01	40.64
Candidate-only calibration	Gap gate	Head+Top (candidate-only)	55.27	32.13	50.34	7.26	37.03	46.70	66.72	50.40	43.23	40.84
Head-only probe	Gap gate	LM head (online)	54.16	32.41	50.24	7.22	36.51	45.78	67.55	49.20	42.88	40.45
Top-only probe	Gap gate	Top two blocks (online)	54.34	32.24	50.46	7.24	36.89	45.52	67.55	49.40	42.96	40.52
Raw-cosine reweighting	Uniform	Concat. cosine (Head+Top)	53.42	33.17	50.17	7.16	36.37	45.13	65.81	49.40	42.58	40.12

Token and group allocation. Token gate only and group alignment only improve TaskAvg/CatAvg over vanilla GKD, reaching 42.68/40.28 and 43.45/41.05, respectively. The full method reaches 44.30/41.81, exceeding either component alone on both aggregates. This aggregate benefit coexists with task-specific differences: group alignment only scores higher on TruthfulQA-MC2 (50.94 versus 50.60).

Calibration of the scoring gradients. To distinguish scoring-gradient calibration from token weighting in the final update, we retain the final token-gated loss and Head+Top rank fusion while varying calibration on the two sides of the gradient comparison. Uncalibrated alignment uses uncalibrated gradients for both candidate groups and the controller. Candidate-only calibration applies the token gate to candidate scoring gradients while leaving the controller scoring gradient uncalibrated. Both variants retain the controller protocol and rank-to-weight mapping of the full method, including the group-weight range of $[0.8, 1.2]$. Calibrating both sides improves TaskAvg/CatAvg by 1.29/1.18 points over uncalibrated alignment and by 1.07/0.97 points over candidate-only calibration. These comparisons support calibration-aware allocation: token calibration informs both the final update loss and the gradient comparisons used to assign group weights.

Gradient spaces and score fusion. Head-only and top-only probes yield lower aggregates than the combined Head+Top probe. Raw-cosine reweighting uses uniform token weights and ranks a single cosine computed from concatenated, uncalibrated Head+Top gradients. Group alignment only, which ranks the two spaces separately and fuses their ranks, exceeds raw-cosine reweighting by 0.87/0.93 TaskAvg/CatAvg points, with gains on seven of eight tasks. The local interventions below examine how allocation scores relate to the benefits of individual updates.

4.4 Local Effects of Supervision Allocation

We study one-step AdamW updates at steps 300, 600, 900, and 1,200 of a single vanilla GKD trajectory for Qwen2.5-7B-to-1.5B distillation. Comparisons start from the same model and optimizer state and reuse the same generated responses. We measure outcomes using the base GJS loss on an independent probe, averaged over response tokens without token or group weighting. Appendix A.9 gives the sampling and update protocol, with predictor definitions in Appendix A.9.1.

To test whether group scores predict useful training contributions, we hold Token Gate fixed and increase one group's weight from 1 to 1.2 while reducing the other seven weights so that the eight weights still average one. Allocation utility is the probe-loss reduction relative to an update with uniform group weights. Head+Top has the largest macro Spearman correlation point estimate among the tested signals, $\rho = 0.50$ (Table 4A). The two highest-scoring groups exceed the two lowest-scoring groups in allocation utility by 2.95×10^{-5} on average, with a 95% confidence interval of $[2.20, 3.70] \times 10^{-5}$.

Table 4: **Local allocation utility at four GKD checkpoints.** Macro Spearman correlations (A) and one-step AdamW gains (B). Gains are control loss minus method loss ($\times 10^{-5}$); brackets give 95% paired bootstrap confidence intervals. Checkpoints receive equal weight.

A. Group utility prediction		
Predictor	Macro ρ	95% CI
Head+Top rank	**0.50**	[0.41, 0.58]
Raw gradient cosine	0.31	[0.21, 0.41]
Positive CE gap	0.19	[0.08, 0.30]
Candidate JSD	0.14	[0.03, 0.25]
Gradient norm	0.09	[−0.03, 0.21]
Static response length	−0.03	[−0.14, 0.09]
Shuffled score	0.01	[−0.10, 0.11]
B. AdamW probe-loss gains ($\times 10^{-5}$)		
Contrast	Macro gain	95% CI
Group: Attune vs. Uniform	+1.95	[1.35, 2.55]
Group: Attune vs. Shuffle	+1.70	[1.05, 2.35]
Token: Gate vs. Uniform	+1.40	[0.90, 1.90]
Token: Gate vs. Shuffle	+1.25	[0.75, 1.75]

We also compare the full Attune group allocation with uniform and shuffled group weights. To isolate token calibration, we fix all group weights to one and compare Token Gate with uniform token weights and weights shuffled among valid response tokens within each group. Both components reduce probe loss relative to their uniform and shuffled controls on average (Table 4B). The gains over shuffled controls support the value of assigning weights to specific groups and tokens. These results provide local evidence for the allocation mechanism, complementing the downstream evaluation of the complete training procedure.

5 Related Work

On-policy distillation learns from teacher feedback on student-generated responses (Agarwal et al., 2024). TIP, TA-OPD, and FiRe-OPD refine token or trajectory selection and weighting (Xu et al., 2026b; Wang et al., 2026c; Li et al., 2026). Curriculum learning adapts supervision to difficulty and competence (Bengio et al., 2009; Kumar et al., 2010). DMC and PACED further guide distillation through data–model compatibility and student success rates (Huang et al., 2026; Xu et al., 2026a). Our controlled arithmetic diagnosis complements this perspective by comparing the relative gains from fixed basic and difficult supervision across student capacities and training states under matched token exposure.

Gradient-based approaches connect supervision allocation to reference objectives. Ren et al. (2018) learn example weights from validation feedback; LESS selects influential instruction data (Xia et al., 2024); GradAlign uses gradient alignment for selection in LLM reinforcement learning (Yang et al., 2026). Attune couples soft token calibration with online group allocation during on-policy distillation. It applies the same calibration to candidate and reference losses before comparing their gradients, coordinating token weighting with allocation. The reference is refreshed using teacher feedback on the current student's responses.

6 Conclusion

We present a controlled diagnosis of supervision utility and Attune, a coupled allocation framework for on-policy distillation. Paired SFT interventions in compositional arithmetic show that fixed supervision has capacity- and state-dependent returns under matched token exposure. Attune connects token calibration to online group allocation: calibrated candidate and reference gradients determine how each group is weighted. Three-seed comparisons across two model families show aggregate benchmark gains over vanilla GKD and selected baselines, while independent-probe interventions show larger one-step reductions in distillation loss than uniform and shuffled weighting. Component comparisons support incorporating calibration into allocation decisions, beyond applying token weights only to the final loss. These results favor coordinating how supervision is transformed and allocated as the learner evolves.

Reproducibility Statement

We provide the complete training algorithm in Appendix A.1 and the diagnostic construction, data partitions, and training controls in Appendix A.2. Training and evaluation settings, baseline configurations, and model-FLOP accounting are specified in Appendices A.6, A.7, and A.11, respectively. Appendix A.9 details the local intervention protocol. Code for the main method is provided in the Supplementary Material.

AI Use Statement

We used large language models to assist with language editing and organization, literature search and identification of related work, reference checking and formatting, and LaTeX typesetting. The authors reviewed and verified the AI-assisted manuscript content, including citations and technical claims, and take full responsibility for the final submission.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5be69a584901a26c521c2b51e40a4c20-Abstract-Conference.html.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48, 2009. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

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. doi: 10.48550/arXiv.2110.14168. URL https://arxiv.org/abs/2110.14168.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3029–3051. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.183. URL https://aclanthology.org/2023.emnlp-main.183/.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2368–2378. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1246. URL https://aclanthology.org/N19-1246/.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, July 2024. URL https://zenodo.org/records/12608602.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*, 2024. URL https://arxiv.org/abs/2406.20094.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8ac015d409635f196f9e3e9dcfb9a94e-Abstract-Conference.html.

Changyi He, Yifu Ding, Jinyang Guo, Ruihao Gong, Haotong Qin, and Xianglong Liu. DA-KD: Difficulty-aware knowledge distillation for efficient large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 22379–22391. PMLR, 2025. URL https://proceedings.mlr.press/v267/he25c.html.

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. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531.

Jiahao Huang, Fei Cheng, Junfeng Jiang, and Akiko Aizawa. Tailoring the curriculum: Student-centered reasoning distillation via dynamic data-model compatibility. *arXiv preprint arXiv:2605.29229*, 2026. URL https://arxiv.org/abs/2605.29229.

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. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31044–31062. PMLR, 2025. URL https://proceedings.mlr.press/v267/ko25a.html.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 23, 2010. URL https://proceedings.neurips.cc/paper/2010/hash/e57c6b956a6521b28495f2886ca0977a-Abstract.html.

Hynek Kydlíček. Math-Verify: Math verification library. GitHub software repository, 2025. URL https://github.com/huggingface/Math-Verify.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=i1uGbfHHpH.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. NuminaMath TIR. Hugging Face dataset, 2024. URL https://huggingface.co/datasets/AI-MO/NuminaMath-TIR.

Jijie Li, Li Du, Hanyu Zhao, Bo-wen Zhang, Liangdong Wang, Boyan Gao, Guang Liu, and Yonghua Lin. Infinity instruct: Scaling instruction selection and synthesis to enhance language models. *arXiv preprint arXiv:2506.11116*, 2025. URL https://arxiv.org/abs/2506.11116.

Yuying Li, Leqi Zheng, Yongzi Yu, Wenrui Zhou, Xuchang Zhong, Xing Hu, Jing Jin, Hangjie Yuan, and Tao Feng. Filter, then reweight: Rethinking optimization granularity in on-policy distillation. *arXiv preprint arXiv:2606.02684*, 2026. URL https://arxiv.org/abs/2606.02684.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. The Flan collection: Designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22631–22648. PMLR, 2023. URL https://proceedings.mlr.press/v202/longpre23a.html.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Erran Li, Raluca Ada Popa, and Ion Stoica. DeepScaleR: Surpassing O1-Preview with a 1.5b model by scaling RL. Technical blog, 2025. URL https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2.

Meta. Llama 3.2 model card. Model card, 2024. URL https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md.

Open R1 Team. OpenR1-Math-220k. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/open-r1/OpenR1-Math-220k.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4334–4343. PMLR, 2018. URL https://proceedings.mlr.press/v80/ren18a.html.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. MuSR: Testing the limits of chain-of-thought with multistep soft reasoning. In *International Conference on Learning Representations*, volume 2024, pp. 14670–14728, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/3f8c7eb848ffec848f3ed2b7ca44915d-Abstract-Conference.html.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3621–3634. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-acl.317. URL https://aclanthology.org/2021.findings-acl.317/.

Teknium. OpenHermes 2.5: An open dataset of synthetic data for generalist LLM assistants. Hugging Face dataset, 2023. URL https://huggingface.co/datasets/teknium/OpenHermes-2.5.

Guanghui Wang, Kaiwen Lv Kacuila, Zhiyong Yang, Zitai Wang, Jin-Wen Wu, Longtao Huang, Qianqian Xu, and Qingming Huang. The bridge-garden dilemma in LLM distillation: Why mixing hard and soft labels works. In *International Conference on Machine Learning*, 2026a. URL https://arxiv.org/abs/2605.26246. Accepted at ICML 2026.

Minzheng Wang, Yongbin Li, Haobo Wang, Xinghua Zhang, Nan Xu, Bingli Wu, Fei Huang, Haiyang Yu, and Wenji Mao. Adaptive social learning via mode policy optimization for language agents. In *International Conference on Learning Representations*, 2026b. URL https:

//proceedings.iclr.cc/paper_files/paper/2026/hash/063d1fadbde557466bf5bcffe814a64e-Abstract-Conference.html.

Yuanyi Wang, Su Lu, Yanggan Gu, Pengkai Wang, Yifan Yang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, and Hongxia Yang. Not all disagreement is learnable: Token teachability in on-policy distillation. *arXiv preprint arXiv:2605.26844*, 2026c. URL https://arxiv.org/abs/2605.26844.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, pp. 95266–95290. Curran Associates, Inc., 2024. doi: 10.52202/079017-3018. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/ad236edc564f3e3156e1b2feafb99a24-Abstract-Datasets_and_Benchmarks_Track.html.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54104–54132. PMLR, 2024. URL https://proceedings.mlr.press/v235/xia24c.html.

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 *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a2747a3844ca1e4667fbff3f558eb39b-Abstract-Conference.html.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, and Zhipeng Wang. PACED: Distillation and on-policy self-distillation at the frontier of student competence. *arXiv preprint arXiv:2603.11178*, 2026a. URL https://arxiv.org/abs/2603.11178.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. TIP: Token importance in on-policy distillation. *arXiv preprint arXiv:2604.14084*, 2026b. URL https://arxiv.org/abs/2604.14084.

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*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/be06e3802e9411381feece79b4d960c1-Abstract-Conference.html.

Ningyuan Yang, Weihua Du, Weiwei Sun, Sean Welleck, and Yiming Yang. GradAlign: Gradient-aligned data selection for LLM reinforcement learning. *arXiv preprint arXiv:2602.21492*, 2026. URL https://arxiv.org/abs/2602.21492. Accepted at COLM 2026.

Tian Ye, Zicheng Xu, Yuanzhi Li, and Zeyuan Allen-Zhu. Physics of language models: Part 2.1, grade-school math and the hidden reasoning process. In *International Conference on Learning Representations*, pp. 97699–97709, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f3064f7a0ca2328ecb41a3aef6177d68-Abstract-Conference.html.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. MetaMath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations*, pp. 45040–45061, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/c400474e8a36d0812fdee52739288b12-Abstract-Conference.html.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, pp. 113222–113244, 2025. doi: 10.52202/085713-3775. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. WildChat: 1M ChatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/9421261e06f1a63a352b068f1ac90609-Abstract-Conference.html.

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. doi: 10.48550/arXiv.2311.07911. URL https://arxiv.org/abs/2311.07911.

A Additional Method and Experimental Details

This appendix provides the complete training algorithm, diagnostic protocols, data construction, training and evaluation settings, local update comparisons, smaller-student results, and computational details.

A.1 Complete Attune Training Algorithm

Algorithm 1 expands the training step described in Section 3. Both the candidate and controller losses use the same detached token calibration; controller examples affect training only through the resulting group weights.

Controller sampling. From the 2,000-example controller split, we fix a seeded pool of 512 prompts and take the next 32 cyclically to form the global controller minibatch at each step, covering the pool every 16 steps. The current student regenerates their responses using the training decoding policy and an independent deterministic random-number stream; the teacher scores the same trajectories. The instruction mixture covers instruction following, knowledge, reading, reasoning, and mathematics, with its composition given in Appendix A.6.

Distributed objective. Each training group is one worker-local minibatch of eight trajectories, giving $B = 8$ groups per step on eight data-parallel workers. Each worker forms a local response-token mean, scales it by its detached group weight, and data-parallel reduction averages these contributions as in Equation 5. Normalization sets the mean token weight within each group and the mean group weight across groups to one; it does not fix the loss value or gradient norm.

Numerical safeguards. The reported runs use a KD temperature of 1. Token calibration uses $\epsilon = 10^{-8}$ to floor the raw pace weights and the group-mean denominator. For a nonempty group, the raw-weight floor ensures $\mu_{t,b} \geq \epsilon$, so the denominator in Equation 2 is equivalent to $\max(\mu_{t,b}, \epsilon)$. No clipping is applied after normalization and shrinkage. After normalization, a token's final weight depends on the other gaps in its group; equal gaps yield uniform weights. For Equation 3, dot products and norms are accumulated in FP32, and each gradient norm is floored at 10^{-12} before division; a zero probe gradient therefore yields a zero score.

Relative group weights. For $B > 1$, a cosine with average one-based rank k_b receives $q_b = (k_b - 1)/(B - 1)$. Average ranks sum to $B(B + 1)/2$, including ties, so the percentiles have mean $1/2$. For $B = 1$, we set $q_1 = 1/2$, yielding unit group weight. If all cosines are negative, the least

Algorithm 1 Attune training step

Require: Student π_θ, fixed teacher π_{T}, training distribution $\mathcal{D}$, controller pool $\mathcal{C}$, probe spaces $\Phi = \{\mathrm{head}, \mathrm{top}\}$

```
Sample a global training batch B_t = {G_{t,1}, ..., G_{t,B}} from D
Form the global controller minibatch C_t from the next 32 prompts in the fixed cyclic pool
Generate training and controller responses with the current student
Evaluate the teacher on all student-generated trajectories
Compute base GJS token losses for both streams
Compute detached token pace weights using Equation 2
Form calibrated group losses L^cal_{t,b} and controller loss L^cal_{C_t}
for each probe space r ∈ Φ do
    Compute controller probe gradient g^(r)_{C_t}
    for each training group b = 1, ..., B do
        Compute calibrated group probe gradient g^(r)_{t,b}
        Compute directional compatibility s^(r)_{t,b} using Equation 3
    end for
    Convert the B similarities to percentile ranks q^(r)_{t,b}
end for
Fuse head/top ranks and compute w_{t,b} ∈ [0.8, 1.2] using Equation 4
Form L_Attune using Equation 5
Take one AdamW step on all student parameters using ∇_θ L_Attune
```

Table 5: Data partitions in the controlled arithmetic study. The intervention set comprises one pre-registered Anchor/Hard pair per seed.

Partition	Examples	Role
Calibration	5,000	Difficulty and format calibration
Parent stream	6,144	Single anchor-only pass
Interventions	3,072	Three fixed Anchor/Hard block pairs
Probe	5,000	Held-out free-generation evaluation
Total	19,216	Unique examples in confirmatory protocol

conflicting group can still receive the largest weight: ranks describe relative compatibility within the batch.

A.2 Detailed Protocol for the Controlled Arithmetic Study

This section details the data construction, partitions, and training budgets for the full-parameter SFT diagnosis in Section 2.

Controlled data construction. The arithmetic task uses executable modular arithmetic to vary dependency depth while keeping operation count and target-trace length fixed. Every example contains five additions, five subtractions, and five multiplications modulo 23, arranged into a 15-node expression graph. The target contains one line for every operation. Levels 1–5 have critical-path lengths $4/6/8/11/15$, respectively, while their mean response lengths remain between 431.95 and 432.06 Qwen2.5 tokens. Levels 1–3 form the anchor slice and Levels 4–5 form the hard slice.

The confirmatory protocol uses 19,216 unique examples. The calibration, parent, intervention, and probe partitions have disjoint sample identifiers.

Single-pass intervention design. For each capacity and seed, the parent stream is traversed in a fixed random order exactly once. Global batch size is 64, giving 96 parent optimizer steps. We save parent states at steps $0/24/48/72/96$, corresponding to $0/25/50/75/100\%$ progress. For each training seed s, we pre-register one 512-example anchor block A_s and one 512-example hard block H_s. At every state, two independent copies of the parent checkpoint are trained for eight steps on

A_s or H_s. The same pair is also used for all four capacities under seed s, with sample identities and within-block order held fixed. Intervention pairs are disjoint across seeds, and every independent model branch processes its assigned block exactly once.

All blocks contain approximately 221,200 supervised response tokens. Across the three fixed seed-specific pairs, the Anchor/Hard response-token mismatch is at most 0.03%, and prompt-token mismatch is at most 0.35%.

Models and optimization. We use the official instruction-tuned Qwen2.5 checkpoints at 0.5B, 1.5B, 3B, and 7B parameters (Qwen Team, 2024). Every run performs full-parameter BF16 SFT with maximum sequence length 1,024, maximum prompt length 256, AdamW, weight decay 0.01, gradient clipping at 1.0, gradient checkpointing, no warmup, and a constant learning rate. Learning rates are 5×10^{-6}, 1×10^{-5}, 3×10^{-6}, and 2×10^{-6} for 0.5B, 1.5B, 3B, and 7B, respectively. Per-device batch size and gradient accumulation are selected per capacity while preserving global batch size 64. Seeds 10, 20, and 30 change the fixed parent order, the assigned Anchor/Hard pair, and stochastic training state; all runs begin from the same official checkpoint at each capacity.

Generation and statistics. The held-out probe contains 1,000 examples at each level. The primary analysis evaluates a fixed balanced 500-example subset, with 100 examples per level. We apply the Qwen chat template and greedily generate up to 1,024 new tokens with an answer-aware stopping rule. Final numeric equality modulo 23 is the primary metric. We additionally execute the 15 generated intermediate steps, and record parseability, generated length, and maximum-length hits as diagnostics. Invalid final answers receive zero credit.

For each seed and state, marginal utility is the post-intervention score minus the corresponding parent score. Relative utility is the paired Hard-minus-Anchor score, for which the parent term cancels. State curves report the mean and sample standard deviation over three seeds; the sample SD uses denominator $3 - 1$.

Endpoint 95% confidence intervals use a paired, level-stratified bootstrap over the crossed seed and probe-example axes. Each replicate samples three seed indices with replacement and, independently, 100 probe-example indices with replacement within each of the five difficulty levels. The same sampled example indices are used across all sampled seeds, preserving the Hard/Anchor outcome pair for each seed and example. We first compute each sampled example's Hard-minus-Anchor correctness difference, then average equally over the 100 examples per level, five levels, and three sampled seeds, and convert the result to percentage points. The 2.5th and 97.5th percentiles of this replicate distribution form the intervals in Table 6 and Figure 1(b). Thus SD describes variation across the three observed seeds, while the bootstrap intervals also reflect uncertainty from the finite probe.

Table 6: Endpoint hard-minus-anchor utility in exact-match percentage points. Mean ± SD summarizes three paired seeds; 95% CIs use paired, level-stratified bootstrap resampling.

Student	Mean ± SD	Seed 10 / 20 / 30	95% CI
0.5B	-2.93 ± 0.95	$-2.60/-4.00/-2.20$	$[-5.37, +0.70]$
1.5B	$+7.20 \pm 1.83$	$+5.20/+8.80/+7.60$	$[+3.67, +9.80]$
3B	$+9.73 \pm 1.33$	$+11.20/+8.60/+9.40$	$[+7.03, +12.13]$
7B	$+11.80 \pm 0.40$	$+11.40/+11.80/+12.20$	$[+10.27, +13.83]$

Absolute utility and intermediate-step diagnostics. Figure 2 reports each intervention's absolute accuracy gain over its corresponding parent at 0%, 25%, 50%, 75%, and 100% progress. Point labels show three-seed means, and ribbons show one sample SD across seeds. Each panel uses a capacity-specific vertical scale to make within-student changes legible; Figure 1 compares the relative Hard-minus-Anchor utilities across capacities. The endpoint confidence intervals in Figure 1(b) concern this relative difference.

At 75% and 100% progress, both interventions yield positive mean gains over the parent for every student. Anchor yields the larger gain for 0.5B, while hard supervision yields the larger gain for the larger students. At these same states, hard supervision improves the 0.5B student's intermediate-step

Table 7: Data composition in the three-domain study. Counts are per difficulty level. Response lengths are training-tokenizer means for Levels 1–5; the parenthesized value is the overall median.

Domain	Source	Train/ level	Probe/ level	Level construction	Mean response tokens L1–L5	Answer rule
Controlled arithmetic	iGSM generator	800	200	operations: 2/4/ 7/10/14	45/94/164/231/317 (163)	numeric equality modulo 23
Natural mathematics	OpenR1-Math-220k	800	200	response-complexity quintiles	985/1343/1537/1684/ 1768 (1510)	symbolic equivalence
Deductive logic	ProofWriter OWA depth-5	800	200	proof depth: 0/ 1/2/3/5	25/47/63/80/109 (61)	exact class label
Total	three domains	12,000 train / 3,000 probe		15 balanced cells	–	–

accuracy relative to anchor supervision by $+2.40$ and $+5.95$ points, while the corresponding final-answer utility differences are -3.00 and -2.93 percentage points. Better local execution therefore does not necessarily translate into reliable completion of a long dependency chain.

A.3 Three-Domain Supervision Diagnosis

We complement the controlled arithmetic diagnosis with experiments spanning procedural arithmetic, natural mathematics, and deductive reasoning. Each domain uses its own difficulty criterion, and response lengths vary across levels.

Data construction. The three-domain study uses 12,000 training examples and a disjoint 3,000-example probe. Each of its three domains is divided into five within-domain, construction-defined difficulty levels; every domain–level cell contains 800 training and 200 probe examples. Levels 1–3 form the anchor slice and Levels 4–5 form the hard slice.

Using the iGSM generator (Ye et al., 2025), we set the number of arithmetic operations to $2/4/7/10/14$ for Levels 1–5. For ProofWriter, the open-world-assumption depth-5 release supplies proof depths $0/1/2/3/5$ (Tafjord et al., 2021); the main split is balanced between `True` and `False`. For OpenR1-Math-220k, we retain English-like records passing the released reasoning-completeness and mathematical-correctness filters, deduplicate by problem identifier, and stratify responses into quintiles of response complexity (Open R1 Team, 2025).

Training and probe source identifiers are disjoint. iGSM instances are split at generation time, ProofWriter uses separate source theories, and OpenR1 is split by unique problem ID. Normalized exact matching finds no overlap between the 4,000 OpenR1 training problems and the GSM8K or MATH evaluation problems.

Models and optimization. We use the official instruction-tuned Qwen2.5 checkpoints at 0.5B, 1.5B, 3B, and 7B parameters (Qwen Team, 2024). Every run performs full-parameter BF16 SFT on fixed target responses, with maximum sequence length 2,048, maximum prompt length 384, AdamW, weight decay 0.01, gradient clipping at 1.0, global batch size 64 across eight GPUs, and gradient checkpointing. Learning rates are 5×10^{-6} for 0.5B, 3×10^{-6} for 1.5B and 3B, and 2×10^{-6} for 7B, with 3% linear warmup followed by a constant schedule. Per-device batch size and gradient accumulation are $(8, 1)$ for 0.5B/1.5B, $(4, 2)$ for 3B, and $(2, 4)$ for 7B.

Treatments and budget matching. `r000` uses Levels 1–3; `r050` replaces half of that distribution with Levels 4–5 while preserving example count and domain balance; and `r100` uses Levels 4–5 exclusively.

Hard traces are longer, so equal-step training would expose hard-heavy runs to more target tokens. The primary comparison matches cumulative response tokens to `r000` step 400. Depending on

Table 8: Distribution treatments in the three-domain protocol. Counts in L1–L5 are per domain; token totals include responses but exclude prompts.

Treatment	L1/L2/L3	L4/L5	Examples	Response tokens
r000	800/800/800	0/0	7,200	3,442,504
r050	400/400/400	600/600	7,200	4,226,059
r100	0/0/0	800/800	4,800	3,350,855

Table 9: Token-matched all-hard minus anchor-only change (r100−r000), in accuracy percentage points. Overall averages the balanced five-level probe (60% anchor and 40% hard).

Student	Overall	Anchor L1–3	Hard L4–5	External macro
0.5B	−12.17	−20.56	+0.42	−1.69
1.5B	−5.94	−9.98	+0.11	−0.36
3B	−2.50	−9.07	+7.36	+0.52
7B	−0.10	−6.76	+9.89	+2.48

model and seed, this selects steps 324–326 for r050 and 272–274 for r100; residual mismatch is below 0.74%. Prompt tokens are excluded.

Probe and external evaluation. The primary probe contains 3,000 balanced examples. Greedy free generations are scored by numeric equivalence for iGSM, the released mathematical verifier for OpenR1, and exact final True/False/Unknown classification for ProofWriter. Invalid and unparseable outputs are incorrect. The completed capacity runs used maximum generation lengths of 512 for iGSM and OpenR1 and 256 for ProofWriter. OpenR1 generations frequently reached this 512-token ceiling, so its scores reflect a constrained generation budget.

Transfer evaluation uses all 1,319 GSM8K test questions, a deterministic 500-question MATH subset with 100 official test problems per level, and all 750 questions from the three BBH logical-deduction tasks. Their unweighted macro-average is supplementary and was not used to construct treatments.

Selection and randomness. The 400-step endpoint was selected once using seed 10 and mean probe accuracy over all four capacities for r000 and r050. The all-hard treatment, external benchmarks, and seeds 20 and 30 were excluded from this choice. We then completed a factorial design with four capacities, three treatments, and three seeds (36 trajectories). Seeds change data order and stochastic training operations; each run starts from the same official checkpoint at its capacity. We report the mean and sample standard deviation of paired differences across three seeds.

A.4 Additional Three-Domain Analyses

All-hard stress test. Replacing the anchor distribution entirely with hard examples improves the hard probe slice slightly for the 0.5B and 1.5B students and more substantially for the 3B and 7B students, but incurs an anchor-slice cost at every capacity. The aggregate score falls for every model. The anchor retention cost is measured relative to the anchor-only baseline, r000.

Domain heterogeneity. The clearest capacity-dependent pattern appears on iGSM. OpenR1-Math changes from slightly negative at 0.5B to modestly positive for larger students, whereas ProofWriter remains negative but its loss shrinks with capacity.

NLL–generation mismatch. Mixed training lowers teacher-forced target NLL in all 12 paired comparisons (four capacities and three seeds), but aggregate free-generation accuracy decreases for the two smaller students. This conclusion is unchanged under two NLL aggregations. Token-weighted NLL pools all target tokens and consequently gives more weight to the longer OpenR1-Math responses. Cell-macro NLL first computes token-weighted NLL within each of the 15 domain–level cells and then weights those cells equally. Both are evaluated on the same fixed probe and at the same response-token-matched checkpoints as the generation comparison.

Table 10: Token-matched r050−r000 accuracy change by domain, in percentage points.

Student	iGSM	OpenR1-Math	ProofWriter
0.5B	+4.47	−0.57	−6.90
1.5B	+3.37	+0.53	−5.13
3B	+12.03	+1.47	−3.67
7B	+15.57	+2.43	−0.30

Table 11: Paired token-matched changes under mixed versus anchor-only training (r050−r000). NLL entries are three-seed means ± sample standard deviations; negative values indicate improved teacher-forced fit. "Negative seeds" counts paired seeds for which both NLL aggregations decrease. Accuracy is free-generation exact-match change on the balanced 3,000-example probe, in percentage points.

Student	Δ token NLL	Δ cell-macro NLL	Negative seeds	Δaccuracy
0.5B	-0.0214 ± 0.0013	-0.0380 ± 0.0044	3/3	−1.00
1.5B	-0.0118 ± 0.0004	-0.0237 ± 0.0012	3/3	−0.41
3B	-0.0128 ± 0.0009	-0.0235 ± 0.0011	3/3	+3.28
7B	-0.0118 ± 0.0014	-0.0206 ± 0.0028	3/3	+5.90

All NLL contrasts are computed within each model capacity. These comparisons show that a treatment can improve likelihood under reference prefixes without improving the student's ability to generate a correct answer autoregressively. We report NLL as an optimization diagnostic and free-generation accuracy as the capability endpoint throughout the diagnosis.

A.5 Repeated-Exposure Curriculum Pilot

We also compared a fixed random order with global easy-to-hard (E2H) and hard-to-easy (H2E) orders over one identical 7,168-example multiset. Steps 112, 224, and 336 correspond to one, two, and three complete exposures. The optimizer and constant learning rate are shared, and each training order is reused across exposures. After exploratory seed-10 runs at all capacities, the H2E-minus-random contrast was repeated with three seeds for 1.5B and 3B.

The early internal H2E advantage fades after repeated exposure. In this pilot, training state changes jointly with exposure count, and H2E places anchor examples at the end of each pass, coupling state with recent supervision. External gains vary across capacities and exposure counts. These observations motivate the single-pass comparison in Section 2.3, which applies fixed intervention blocks to independent copies of each parent state.

A.6 Reproducible Experimental Setup

Models and training data. Our primary setting distills Qwen2.5-7B-Instruct into Qwen2.5-1.5B-Instruct (Qwen Team, 2024). The teacher is frozen and all student parameters are updated. For the main comparisons, we construct a mixture of 100,000 English prompts from public instruction-following and mathematics datasets (Table 13), with mutually disjoint splits of 96,000 training, 2,000 controller, and 2,000 development prompts. The mixture contains 22% general instruction following, 25% knowledge, 10% reading comprehension, 23% reasoning, and 20% mathematics. We retain English prompts of 8–340 Qwen2.5 content tokens and apply language, format, and source-specific quality filters. Source responses are used for filtering and static selection, but are not used as training targets for the on-policy methods. The four Tülu 3 subsets are drawn from the released Tülu 3 SFT mixture (Lambert et al., 2025). OpenHermes examples come from records labeled OpenHermes-2.5 in the locally archived qdr91/infinity-instruct-100k release.

Deduplication and overlap screening. Before assigning splits, we remove exact duplicates after Unicode normalization, lowercasing, and removal of non-alphanumeric characters. We also remove near duplicates across sources using 64-bit token SimHash with Hamming distance at most 3 and normalized-text length ratio at least 0.85. During construction, we screen candidates against locally

Table 12: Pilot H2E minus fixed-random effect at matched exposure endpoints, in accuracy percentage points (mean ± sample SD over three paired seeds). "Positive seeds" counts the seeds for which H2E outperforms fixed-random ordering on the internal probe.

Student	Exposures (steps)	Internal probe	Positive seeds	External macro
1.5B	1 (112)	$+1.24 \pm 1.15$	3/3	$+3.33 \pm 3.09$
	2 (224)	$+0.91 \pm 0.22$	3/3	$+0.83 \pm 1.21$
	3 (336)	-0.23 ± 1.64	1/3	$+0.30 \pm 0.63$
3B	1 (112)	$+1.83 \pm 1.13$	3/3	$+1.42 \pm 2.13$
	2 (224)	$+0.59 \pm 1.27$	2/3	$+1.76 \pm 2.39$
	3 (336)	-0.01 ± 0.69	2/3	$+2.61 \pm 2.54$

Table 13: Source composition of the English prompt mixture. Counts cover the training, controller, and development splits together. Source names link to dataset releases; citations credit the corresponding work.

Source	Prompts
Infinity-Instruct (filtered) (Li et al., 2025)	29,000
Magpie (Qwen2.5, filtered) (Xu et al., 2025b)	21,000
UltraChat 200K (Ding et al., 2023)	15,000
OpenHermes 2.5 (Teknium, 2023)	1,000
Tülu 3: WildChat (Zhao et al., 2024)	5,000
Tülu 3: PersonaHub instruction following (Ge et al., 2024)	5,000
Tülu 3: FLAN v2 (Longpre et al., 2023)	4,000
Tülu 3: NuminaMath (Li et al., 2024)	9,000
OpenR1-Math-220k (Open R1 Team, 2025)	7,000
MetaMathQA (unique original questions) (Yu et al., 2024)	2,000
DeepScaleR (Luo et al., 2025)	1,000
DAPO-Math (Yu et al., 2025)	1,000
Total	100,000

available evaluation and diagnostic data using exact matching and the same SimHash criterion with a length-ratio threshold of 0.82. This index contains 50,759 unique normalized prompt strings from 232 files, including local BBH, GSM8K, and MATH test data. The construction manifest records the coverage of this screening; it is limited to the indexed evaluation and diagnostic files.

Training-seed comparisons. The full baseline and component comparisons use training seed 10. Table 15 uses training seeds 10, 20, and 30 for each method in both model families: Attune, vanilla GKD, and the selected method baseline (TA-OPD for Qwen and DistiLLM-2 for Llama). Within each model family, we pair Attune and baseline runs with the same training seed and subtract the baseline score from the Attune score on each task and aggregate. TaskAvg and CatAvg are computed within each run before summarizing across runs. The reported gains average the three paired differences; paired-t confidence intervals and tests use these three differences as the observations. Holm correction is applied separately within each model family to 20 comparisons: eight tasks and two aggregates against each of two baselines. Evaluation seed 42 is fixed across runs. The measurement seeds used in Appendix A.9 index local probe sampling and are separate from these repeated training runs.

The largest mean task-level gains over TA-OPD occur on GSM8K, BBH, and MATH-500; for Llama, the largest gains over DistiLLM-2 occur on GSM8K, MMLU-Pro, and TruthfulQA-MC2. Task-level effects vary: six of eight task-level comparisons with the selected method baseline show significant improvements after correction in each family, while Llama MuSR shows no clear advantage over DistiLLM-2.

Llama configuration. We distill Llama-3.2-3B-Instruct into Llama-3.2-1B-Instruct (Meta, 2024) using BF16 training with AdamW and cosine learning-rate decay from 10^{-5} to 10^{-7} without

Table 14: Complete training configuration for the primary distillation setting.

Setting	Value
Teacher checkpoint	Qwen2.5-7B-Instruct (frozen)
Student checkpoint	Qwen2.5-1.5B-Instruct
Training prompts	96,000
Controller split / fixed pool	2,000 / 512 prompts
Per-step controller prompts	32 (global)
Development prompts	2,000
Optimizer	AdamW
Peak / final learning rate	10^{-5} / 10^{-7}
Learning-rate schedule	Cosine decay
Warmup steps	0
Optimizer steps	1,500
Global prompt batch size	64 (8 per GPU)
Data-parallel training groups	8 groups of 8 prompts
Gradient accumulation	1
Maximum prompt length	384 tokens
Maximum generated response length	8,192 tokens
Rollout temperature	0.9
Rollout top-p	1.0
Rollout top-k	0 (disabled)
GJS coefficient β	0.5
Token-gate function	$r = \max(\exp(-0.25d), 10^{-8})$
Token-weight calibration	$a = 0.75 + 0.25r/\mu$, with group mean μ
Post-normalization token clipping	None
Group strength / range	$\lambda_{\text{grp}} = 0.2$; $[0.8, 1.2]$
Controller refresh interval	Every optimizer step
Gradient-probe spaces	LM head + final two blocks (0.7/0.3)
Numerical precision	BF16 training; FP32 probe accumulation
Weight decay	10^{-2}
Gradient clipping	1.0
Hardware	8 × NVIDIA RTX6000 Blackwell
Main comparison seed	10

warmup. The data, token-length limits, and remaining training settings follow the Qwen configuration in Table 14.

Evaluation protocol. The eight benchmarks are evaluated using their task-specific metrics. Dataset versions, splits, prompts, decoding settings, and extraction rules are frozen before the final method comparison. Both model families are evaluated with `lm-evaluation-harness` (Gao et al., 2024) and evaluation seed 42. MATH-500 uses the 500-problem MATH test subset released with PRM800K (Hendrycks et al., 2021; Lightman et al., 2024), with symbolic answer verification by `Math-Verify` (Kydlíček, 2025). Qwen and Llama use the chat templates supplied with their respective official instruction-tuned tokenizers. Dataset revisions, local-file digests, and task-YAML hashes are stored with the evaluation artifact.

All task metrics are converted to a 0–100 scale before aggregation. *TaskAvg* is the unweighted mean of the eight task scores. *CatAvg* gives equal weight to five capability groups:

$$\text{CatAvg} = \frac{1}{5}\left[s_{\text{IFEval}} + \frac{s_{\text{MMLU-Pro}} + s_{\text{TQA}}}{2} + s_{\text{DROP}} + \frac{s_{\text{MuSR}} + s_{\text{BBH}}}{2} + \frac{s_{\text{GSM8K}} + s_{\text{MATH500}}}{2}\right]. \tag{6}$$

Revision prefixes identify the locally pinned dataset commits. The component comparison in Section 4.3 reports all eight tasks and both aggregates. Its aggregates are computed from the displayed task scores and rounded to two decimal places after aggregation.

Table 15: **Three-seed summaries and paired comparisons on eight benchmarks.** Attune scores are mean ± SD over three training seeds (evaluation seed 42). Δ averages the three training-run differences, Attune minus the paired baseline, in score points. The references are vanilla GKD and selected method baselines: TA-OPD for Qwen and DistiLLM-2 for Llama. CIs are pointwise, unadjusted paired-t 95% intervals (df=2); p_{H} denotes Holm-adjusted p values over 20 tests per family.

(a) Qwen2.5: 7B → 1.5B

	Attune	vs. Vanilla GKD			vs. TA-OPD		
Task	Mean ± SD	Δ	95% CI	p_{H}	Δ	95% CI	p_{H}
IFEval	56.38 ± 0.37	$+3.20$	$[2.67, 3.73]$	0.019	$+1.11$	$[0.65, 1.57]$	0.037
MMLU-Pro	33.61 ± 0.12	$+1.74$	$[1.58, 1.90]$	0.009	$+1.20$	$[0.76, 1.64]$	0.036
TQA-MC2	50.39 ± 0.20	$+0.46$	$[0.40, 0.52]$	0.015	$+0.37$	$[0.29, 0.44]$	0.024
DROP	7.29 ± 0.05	$+0.17$	$[0.01, 0.32]$	0.089	0.08	$[-0.01, 0.17]$	0.089
MuSR	37.43 ± 0.35	$+1.76$	$[1.27, 2.26]$	0.025	$+0.71$	$[0.22, 1.19]$	0.074
BBH	46.66 ± 0.18	$+1.71$	$[1.53, 1.90]$	0.010	$+2.24$	$[1.70, 2.79]$	0.025
GSM8K	70.56 ± 0.22	$+4.65$	$[4.36, 4.94]$	0.004	$+4.65$	$[4.22, 5.08]$	0.009
MATH-500	51.00 ± 0.40	$+2.60$	$[2.10, 3.10]$	0.024	$+2.00$	$[1.50, 2.50]$	0.025
TaskAvg	44.17 ± 0.14	$+2.04$	$[1.82, 2.25]$	0.010	$+1.54$	$[1.25, 1.84]$	0.024
CatAvg	41.70 ± 0.14	$+1.97$	$[1.73, 2.20]$	0.012	$+1.35$	$[1.07, 1.64]$	0.024

(b) Llama-3.2: 3B → 1B

	Attune	vs. Vanilla GKD			vs. DistiLLM-2		
Task	Mean ± SD	Δ	95% CI	p_{H}	Δ	95% CI	p_{H}
IFEval	50.96 ± 0.43	$+2.40$	$[1.94, 2.86]$	0.017	$+2.16$	$[1.89, 2.42]$	0.012
MMLU-Pro	19.88 ± 0.08	$+3.34$	$[3.26, 3.41]$	< 0.001	$+2.71$	$[2.28, 3.15]$	0.017
TQA-MC2	43.38 ± 0.19	$+1.76$	$[1.68, 1.84]$	0.002	$+2.52$	$[2.16, 2.88]$	0.015
DROP	6.65 ± 0.04	$+0.17$	$[0.14, 0.21]$	0.018	$+0.44$	$[0.27, 0.61]$	0.040
MuSR	30.12 ± 0.41	$+0.82$	$[0.59, 1.06]$	0.026	-0.08	$[-0.91, 0.74]$	0.707
BBH	41.50 ± 0.25	$+2.52$	$[2.15, 2.88]$	0.015	$+0.77$	$[0.42, 1.12]$	0.044
GSM8K	49.13 ± 0.35	$+5.05$	$[4.66, 5.45]$	0.006	$+3.21$	$[2.60, 3.81]$	0.017
MATH-500	26.33 ± 0.23	$+0.53$	$[-0.04, 1.11]$	0.114	$+0.60$	$[0.10, 1.10]$	0.105
TaskAvg	33.49 ± 0.11	$+2.08$	$[1.89, 2.26]$	0.007	$+1.54$	$[1.26, 1.82]$	0.017
CatAvg	32.55 ± 0.13	$+1.92$	$[1.73, 2.11]$	0.008	$+1.49$	$[1.24, 1.74]$	0.017

A.7 Baselines and Comparison Setup

The Qwen2.5-7B-to-1.5B and Llama-3.2-3B-to-1B experiments run on eight NVIDIA RTX6000 Blackwell GPUs. Both model families are evaluated under the eight-benchmark protocol described above. Table 17 summarizes the methods' trajectory sources, objectives, and allocation mechanisms; method-specific settings are described below.

FKL, RKL, and Bridge-Garden are reimplementations on the common GKD backbone. DistiLLM-2 (Ko et al., 2025) uses a contrastive objective with skew forward KL on teacher-generated responses and skew reverse KL on student-generated responses. DistiLLM-2 and SKD retain their method-specific response construction. All reported scores come from our evaluations using the task protocols in Table 16.

Configuration and checkpoint selection. For all trained methods in the Qwen and Llama benchmark comparisons, we evaluate the final checkpoint at the end of each training run. This endpoint rule also applies to component variants and repeated-seed runs; the development split is not used to select intermediate checkpoints. For the main Qwen comparisons implemented on the common GKD backbone, we use fixed method-specific configurations and evaluate the 1,500-step endpoint. The training-step comparisons in Appendix A.11 additionally use 1,200-step and 1,900-step endpoints. The default Attune parameters are listed in Table 14. TIP, TA-OPD, and FiRe-OPD use settings documented in their original papers and releases: TIP retains 50% of tokens with Soft-OR scoring and clips entropy at the 98th percentile; TA-OPD uses $K = 16$, a 10% token budget, and

Table 16: Evaluation protocol for the eight benchmarks.

Benchmark	Capability	Version/split	Shots	Decoding	Metric	Direction	Extraction rule
IFEval	Instruction following	`966cd895`; train (541)	0	Greedy	Prompt-level strict accuracy	↑	Official verifier
MMLU-Pro	Knowledge/ reasoning	`b189ec76`; test (12,032)	5	Greedy	Accuracy	↑	Answer-letter extraction
TruthfulQA-MC2	Truthfulness	`741b8276`; validation (817)	0	Choice likelihood	MC2	↑	Official multiple-choice score
DROP	Reading/discrete reasoning	`619386fc`; validation (9,536)	0	Greedy	Token-level F1	↑	Official normalization
MuSR	Multi-step reasoning	`b1f4d416`; author test sets (756)	0	Choice likelihood	Length-normalized accuracy	↑	Answer-choice likelihood
BBH	Compositional reasoning	`982bb89`; test (6,511)	3	Greedy	Macro exact match	↑	Final-answer regex
GSM8K	Mathematics	`740312ad`; test (1,319)	5	Greedy	Exact match	↑	Flexible numeric extraction
MATH-500	Mathematics	`7ecc7947`; test (500)	4	Greedy	Symbolic exact match	↑	`math_verify`

Table 17: Trajectory sources, distillation objectives, and allocation mechanisms of the compared methods.

Method	Trajectory source	Base objective	Token allocation	Group allocation
Official student	–	–	–	–
Teacher-response SFT	Fixed teacher responses	CE	Uniform	Uniform
Vanilla GKD	Student samples	GJS	Uniform	Uniform
On-policy FKL	Student samples	Forward KL	Uniform	Uniform
On-policy RKL	Student samples	Reverse KL	Uniform	Uniform
Bridge-Garden Hybrid KD	Student samples	0.1 FKL + 0.9 teacher-argmax CE	Uniform	Uniform
DistiLLM-2	Teacher/student responses	Contrastive SKL/SRKL	Uniform	Uniform
SKD (top-25)	Interleaved student/teacher	Forward KL	Verification in generation	Uniform
TIP (Soft-OR, 50%)	Student samples	Reverse KL	Hard entropy–divergence selection	Uniform
TA-OPD (K=16, 10%)	Student samples	Reverse KL	Hard teachability selection	Uniform
FiRe-OPD	Student samples	Reverse KL	Entropy soft reweighting	Bottom-20% trajectory filtering
Attune	Student samples	GJS	Soft gap gate	Online controller alignment

5th/95th-percentile normalization; FiRe-OPD filters the bottom 20% of trajectories and sets both token-weight coefficients to 1. SKD uses its top-25 setting, and Bridge-Garden uses the 0.1/0.9 objective mixture in Table 17.

A.8 Weight Range and Refresh Frequency

The default configuration regenerates controller responses and refreshes both probe gradients at every step and maps fused ranks to the bounded range $[0.8, 1.2]$. These choices are fixed across the reported comparisons.

A.9 Local Update Comparisons

Shared setup. The comparisons in Section 4.4 use a Qwen2.5-7B-Instruct teacher and Qwen2.5-1.5B-Instruct student at steps 300, 600, 900, and 1,200 of one vanilla GKD training trajectory. For each of three measurement seeds (101, 202, and 303), we sample four batches of 64 prompts, each partitioned into eight groups of eight. Prompt identities and group assignments are shared across checkpoints, while responses are generated by the student at each checkpoint. This gives 48 state–batch observations and 384 group observations. Each batch has 32 separate controller prompts, and each measurement seed has a fixed set of 128 probe prompts. Candidate, controller, and probe prompts are mutually disjoint.

Within each comparison, all branches reuse the generated responses, teacher distributions, and valid response-token masks. Each branch starts from the same model parameters, AdamW optimizer state, learning rate, and random state. The evaluation loss L_P is the base GJS loss on the independent probe, averaged over valid response tokens with no token or group weighting. After each update, we recompute student distributions on the fixed probe responses while keeping the teacher distributions unchanged.

Group allocation. We keep the default Token Gate fixed and compute a mean gradient for each of the eight groups. The uniform branch assigns every group weight 1. To measure the utility of allocating more weight to group b, we set its weight to 1.2 and each other weight to $1 - 0.2/7$, preserving mean weight 1. Each branch takes one AdamW step using the mean of the weighted group gradients. We define

$$U_b = L_P(\theta_{\text{uniform}}) - L_P(\theta_{\text{boost}\,b}), \tag{7}$$

so positive utility means that increasing the group's allocation improves the update relative to uniform allocation. Within each batch, we compute Spearman correlations between the eight group scores and their utilities, and the mean utility difference between the two highest- and two lowest-scoring groups. We also evaluate the full Attune allocation: Head/Top cosine percentile ranks are fused with weights 0.7/0.3 and mapped to $[0.8, 1.2]$. Its controls use uniform weights or three fixed random permutations of the same Attune weights across groups.

Token calibration. We fix all group weights to 1 and compare uniform token weights, the default Token Gate ($\gamma = 0.25$, $\lambda_{\text{tok}} = 0.25$, with groupwise normalization), and three fixed shuffles of the gate weights among valid response tokens within each group. Shuffling preserves the weight values and their distribution. All branches use the same full batch and take one AdamW step. Gains are the control branch's probe loss minus the gate branch's probe loss, so positive values favor token calibration.

Aggregation and uncertainty. We average the three shuffled-control losses within each state–batch before forming paired contrasts. We then average batch-level correlations and gains within each checkpoint and give the four checkpoints equal weight. The 95% percentile bootstrap intervals resample complete batches within each measurement seed, preserving branch and checkpoint correspondence. The 48 state–batch observations are repeated measurements of 12 candidate batches; the intervals describe variation conditional on this training trajectory and the fixed probe sets. The measurement seeds do not represent independent training runs.

A.9.1 Predictor definitions

We specify the group predictors in Table 4A separately from the allocation utilities they predict. Scores are computed at the parent checkpoint, before any branch update, using the fixed student-generated responses and teacher distributions described above. The static response-length predictor instead uses the fixed source responses paired with these prompts. The independent outcome probe is used only to measure U_b in Equation 7. For a group G_b, let $\mathcal{M}_b$ contain its valid response-token positions and let $N_b = |\mathcal{M}_b|$; prompt and padding positions are excluded. The controller is the common 32-prompt reference minibatch for the same state–batch. For each predictor, a higher score is hypothesized to indicate greater allocation utility; score signs are not chosen from the observed correlations.

Base loss and gradient notation. At KD temperature 1, the base token loss is the symmetric Jensen–Shannon divergence,

$$\begin{aligned} m_{i,u} &= \tfrac{1}{2}(p^{\mathrm{T}}_{i,u} + p^{\mathrm{S}}_{i,u}), \\ J_{i,u} &= \tfrac{1}{2} D_{\mathrm{KL}}(p^{\mathrm{T}}_{i,u} \| m_{i,u}) + \tfrac{1}{2} D_{\mathrm{KL}}(p^{\mathrm{S}}_{i,u} \| m_{i,u}), \\ \mathcal{L}^0(X) &= \frac{1}{N_X} \sum_{(i,u)\in\mathcal{M}_X} J_{i,u}. \end{aligned} \tag{8}$$

KL sums over the full next-token vocabulary and uses natural logarithms. The uncalibrated loss $\mathcal{L}^0$ has no Token Gate or group weights. The calibrated loss $\mathcal{L}^{\text{cal}}$ applies Equation 2, independently within the candidate group or the controller minibatch. For a parameter set $\mathcal{P}$, write $g_X^{0,\mathcal{P}} = \nabla_{\theta_\mathcal{P}} \mathcal{L}^0(X)$ and $g_X^{\text{cal},\mathcal{P}} = \nabla_{\theta_\mathcal{P}} \mathcal{L}^{\text{cal}}(X)$. These are loss gradients before clipping, weight decay, or the AdamW transformation; generated tokens and calibration weights are detached. The head space contains the output projection (`lm_head`); the top space contains all trainable attention, MLP, and normalization parameters within the last two Transformer blocks, excluding the final normalization outside those blocks. Each parameter set contains distinct trainable tensors, counting shared parameters once. Response-token reductions, gradient inner products, and norms are accumulated in FP32.

Head+Top rank. Both candidate and controller gradients are calibrated. We compute a cosine in the LM-head space and a separate cosine in the space of the final two Transformer blocks using Equation 3. Within each space, the eight candidate groups in the same state–batch receive ascending average ranks $k_b^{(r)}$, with ties assigned their mean rank. The predictor is the fused percentile score

$$s_b^{\mathrm{HT}} = 0.7\frac{k_b^{(\mathrm{head})} - 1}{7} + 0.3\frac{k_b^{(\mathrm{top})} - 1}{7}, \tag{9}$$

as in Equation 4. Ranking is not pooled across batches, checkpoints, or measurement seeds. The corresponding group weight $w_b = 0.8 + 0.4 s_b^{\mathrm{HT}}$ is a strictly increasing transform and therefore gives the same Spearman correlation.

Raw gradient cosine. We use uncalibrated candidate and controller gradients in the same LM-head and final-two-block parameter spaces as the method. Following the raw-cosine component baseline in Section 4.3, we flatten and concatenate the spaces before computing a single cosine:

$$\begin{aligned} \widetilde{g}_X^0 &= \big[\,\mathrm{vec}(g_X^{0,\mathrm{head}}); \mathrm{vec}(g_X^{0,\mathrm{top}})\big], \\ s_b^{\cos} &= \frac{\langle \widetilde{g}_{G_b}^0, \widetilde{g}_C^0 \rangle}{\max(\|\widetilde{g}_{G_b}^0\|_2, 10^{-12}) \max(\|\widetilde{g}_C^0\|_2, 10^{-12})}. \end{aligned} \tag{10}$$

The two spaces are concatenated before computing one cosine, without separate normalization, percentile ranks, or 0.7/0.3 mixing. Dot products and norms use the FP32 safeguards in Appendix A.1.

Positive CE gap. On a realized student token, the student-minus-teacher negative-log-likelihood gap is $z_{i,u} = (-\log p_{i,u}^{\mathrm{S}}(y_{i,u})) - (-\log p_{i,u}^{\mathrm{T}}(y_{i,u})) = \log p_{i,u}^{\mathrm{T}}(y_{i,u}) - \log p_{i,u}^{\mathrm{S}}(y_{i,u})$. A positive value means that the teacher assigns higher probability to that realized token. We use the mean positive gap, applying the positive part at each token before aggregation:

$$s_b^{\mathrm{CE}} = \frac{1}{N_b} \sum_{(i,u)\in\mathcal{M}_b} \big[\log p_{i,u}^{\mathrm{T}}(y_{i,u}) - \log p_{i,u}^{\mathrm{S}}(y_{i,u})\big]_+. \tag{11}$$

This unweighted response-token mean summarizes the same discrepancy $d_{i,u}$ used by the Token Gate. It applies neither the exponential pace mapping nor normalized token weights, and negative gaps cannot cancel positive gaps from other positions.

Candidate JSD. The unweighted candidate-divergence score is $s_b^{\mathrm{JSD}} = \mathcal{L}^0(G_b)$ from Equation 8. It averages full-distribution token JSD over all valid response positions in the candidate group, with no Token Gate, group weight, or controller subtraction. This is a token mean, not the equal-weight mean of eight per-response means; longer responses contribute more positions. Larger scores indicate greater teacher–student distributional disagreement, not necessarily greater realized utility.

Gradient norm. We measure the Euclidean norm of the calibrated candidate-group gradient over all trainable student parameters, denoted by $\mathcal{P}_{\mathrm{all}}$:

$$s_b^{\mathrm{norm}} = \|g_{G_b}^{\mathrm{cal},\mathcal{P}_{\mathrm{all}}}\|_2 = \left(\sum_{p\in\mathcal{P}_{\mathrm{all}}} \sum_j \big(g_{G_b,p,j}^{\mathrm{cal}}\big)^2\right)^{1/2}. \tag{12}$$

This matches the candidate gradient used in the group-allocation intervention before multiplication by a group weight. Using the calibrated gradient tests whether its magnitude alone predicts utility, while Head+Top assesses its directional compatibility with the controller. The score includes all layers and embeddings, not only the probe spaces. It is the norm of the response-token-mean loss gradient, not the mean of per-example norms or the norm of an AdamW parameter displacement.

Static response length. We operationalize static complexity as a response-length proxy. For each candidate prompt, let y_i^{src} be the source assistant response paired with the retained prompt after dataset preprocessing. We tokenize its content with the Qwen2.5-1.5B-Instruct student tokenizer,

with special tokens disabled and no length truncation; neither prompt text nor chat-template markers are included. Writing T for this tokenization, the group score is

$$s_b^{\text{length}} = \frac{1}{|G_b|} \sum_{i \in G_b} |T(y_i^{\text{src}})|, \qquad |G_b| = 8. \tag{13}$$

The eight response lengths receive equal weight, with no logarithmic transform or length weighting. Source responses are fixed across the four checkpoints, so this score remains unchanged for a paired candidate group. They supply only the static score and are not used as on-policy training targets. Length measures textual extent rather than intrinsic reasoning difficulty or learnability; Table 4 therefore names this control explicitly as static response length. If a source response is unavailable, the group's static score is marked unavailable rather than filled with a newly generated response.

Shuffled score. We use three uniformly sampled permutations of the fused Head+Top scores across the eight groups of the same state–batch:

$$s_{b,r}^{\text{shuffle}} = s_{\pi_r(b)}^{\text{HT}}, \qquad r \in \{0, 1, 2\}, \tag{14}$$

where the permutations are independent of the measured utilities. For measurement seed $s \in \{101, 202, 303\}$, candidate-batch index $j \in \{0, 1, 2, 3\}$, and repetition r, a dedicated NumPy PCG64 generator initialized by `SeedSequence([s,j,r,1])` generates one permutation of indices 0 through 7. The same index permutation is reused at all four checkpoints of a paired batch, while the Head+Top values are recomputed at each checkpoint. The utility vector remains in its original group order, and candidate groups are indexed by their fixed batch assignment. This preserves the eight score values while breaking their assignment to candidate groups; it does not shuffle tokens, individual probe spaces, or scores across checkpoints. We compute a Spearman correlation for each permutation and average the three correlations within the state–batch before the checkpoint and macro summaries. We do not average shuffled score vectors before computing correlations or treat permutations as independent observations. This score-only control requires no additional AdamW update; the shuffled-weight update controls in panel B measure a separate outcome.

Ties and unavailable scores. Spearman correlations use average ranks for ties in both the predictor and utility. All predictors are compared with the same gated allocation-utility vector from Equation 7. A state–batch with a constant predictor or constant utility has an undefined correlation and is excluded from the affected predictor's correlation summary rather than assigned correlation zero. An empty-response candidate group invalidates the state–batch comparison; a missing source response makes only the static predictor unavailable for that batch. All seven predictors have valid correlations for all 12 candidate batches at each of the four checkpoints; no state–batch observations are excluded. Each predictor's macro correlation therefore averages 48 valid state–batch correlations, with equal weight assigned to the four checkpoint means. These are repeated measurements of the same 12 candidate batches, as accounted for by the paired bootstrap above.

A.10 Additional Qwen2.5-0.5B Results

We additionally evaluate distillation from Qwen2.5-7B-Instruct to Qwen2.5-0.5B-Instruct to examine performance with a smaller student. Table 18 compares Attune with the official student and the distillation baselines using the eight-benchmark evaluation protocol and evaluation seed 42 from Section 4.1.

Attune leads all eight tasks and reaches 29.03 TaskAvg and 27.92 CatAvg, compared with 23.95/22.31 for the official student and 27.18/25.93 for DistiLLM-2, the strongest aggregate baseline.

A.11 Computational Requirements

Relative to vanilla GKD, Attune evaluates one additional controller minibatch and computes gradients in the selected probe spaces during training. The 32-example global controller minibatch is sharded across the eight data-parallel workers. The student architecture is unchanged, and the method adds no parameters or computation at inference.

Table 18: Distillation results for Qwen2.5-7B-to-0.5B on eight benchmarks (%). All metrics are higher-is-better and use a 0–100 scale. † marks separate reference checkpoints; trained rows use seed 10. *TaskAvg* and *CatAvg* follow the definitions in Section 4.1. Bold marks the best result in each column and underlining marks the second-best; ties receive the same formatting.

Method	IFEval	MMLU-Pro	TQA-MC2	DROP	MuSR	BBH	GSM8K	MATH-500	TaskAvg	CatAvg
Anchor										
Official 0.5B student	26.43	18.66	40.52	5.07	30.73	11.72	32.83	25.60	23.95	22.31
Teacher-response SFT (SeqKD)†	33.27	20.32	40.85	5.36	30.92	11.32	31.92	24.00	24.75	23.66
Vanilla GKD†	37.15	19.48	41.26	5.56	32.60	15.36	34.72	27.00	26.64	25.58
Objective and trajectory alternatives										
On-policy FKL†	34.57	20.66	41.83	5.58	30.39	16.00	34.04	26.60	26.21	24.98
On-policy RKL†	33.27	20.46	42.40	5.58	<u>33.47</u>	14.77	31.92	27.20	26.13	24.79
Bridge-Garden Hybrid KD†	34.94	17.94	41.67	5.80	32.52	14.19	34.27	26.60	25.99	24.87
DistiLLM-2†	36.78	21.01	<u>42.57</u>	5.02	32.13	<u>16.62</u>	<u>35.94</u>	27.40	<u>27.18</u>	<u>25.93</u>
SKD (top-25)†	<u>37.71</u>	20.11	41.64	5.03	31.73	15.50	35.56	27.60	26.86	25.76
Student-aware allocation methods										
TIP (Soft-OR, 50%)†	36.60	22.30	41.45	<u>5.89</u>	32.58	15.58	34.50	<u>27.80</u>	27.09	25.92
TA-OPD (K=16, 10%)†	36.78	<u>22.39</u>	42.14	5.68	31.66	16.48	34.87	26.40	27.05	25.89
FiRe-OPD†	37.52	21.48	42.11	5.15	31.64	15.23	35.56	27.60	27.04	25.90
Attune	**41.04**	**22.52**	**44.06**	**5.92**	**34.39**	**17.74**	**37.00**	**29.60**	**29.03**	**27.92**

Model-FLOP accounting. We count model matrix multiplications and attention operations for Qwen2.5-7B-Instruct-to-1.5B-Instruct distillation on eight NVIDIA RTX6000 Blackwell GPUs, with a global training batch of 64 and, for Attune, a global controller minibatch of 32. Both methods use dynamic padding and KV caching during generation. Attune additionally caches the frozen teacher's prefix computations for fixed controller prompts; controller responses are freshly generated and teacher-scored at each step. The sampled mean for Attune is 0.631 PFLOPs per optimizer step. Over 1,500 optimizer steps, including prefix-cache construction, its total model FLOPs are $1.242\times$ those of vanilla GKD.

Training-step budget comparisons. Table 19 reports mean $\pm$ sample SD over three training seeds for the standard 1,500-step runs, an extended GKD run (Long GKD), and a shortened Attune run. Using the model-FLOP ratio above as an approximate guide, the full Attune budget corresponds to $1{,}500 \times 1.242 = 1{,}863$ GKD steps; we round up to the next hundred and run Long GKD for 1,900 steps. At a global batch size of 64, Long GKD uses 121,600 unique training prompts, adding 25,600 previously unused prompts sampled proportionally from the same source datasets. Each ordinary training prompt is exposed only once. In the other direction, $1{,}500/1.242 \approx 1{,}208$ Attune steps approximate the standard GKD budget; we round down to 1,200 steps. The shortened Attune run uses the full run's initialization, training-prompt pool, controller pool, global batch size, and evaluation protocol, with fewer optimizer steps. These budget comparisons use the sampled model-FLOP ratio as an approximation; actual costs depend on generated sequence lengths.

Table 19: **Training-step budget comparisons for Qwen2.5 7B-to-1.5B.** Scores are mean ± sample SD over three training seeds. The full-run Attune scores match Table 2. Training prompts count unique examples used in ordinary training and exclude controller prompts.

Method	Steps	Training prompts	TaskAvg	CatAvg
Vanilla GKD	1,500	96,000	42.13 ± 0.19	39.73 ± 0.21
Long GKD	1,900	121,600	42.55 ± 0.18	39.98 ± 0.11
Attune (shortened run)	1,200	76,800	43.71 ± 0.14	41.23 ± 0.22
Attune (full run)	1,500	96,000	$\mathbf{44.17 \pm 0.14}$	$\mathbf{41.70 \pm 0.14}$

Results across training seeds. Extending GKD from 1,500 to 1,900 steps raises mean TaskAvg/CatAvg from 42.13/39.73 to 42.55/39.98. The 1,200-step Attune run reaches 43.71/41.23, exceeding standard GKD by 1.58/1.50 points and Long GKD by 1.16/1.25 points. The full Attune run reaches 44.17/41.70, retaining gains of 1.62/1.72 points over Long GKD. These comparisons show that extending GKD with additional unique training data does not close the aggregate performance gap in this setting.