SoFT: Soft Targets for Generalizable LLM Fine-Tuning

Anonymous authors
Paper under double-blind review

Abstract

Distillation enables student language models to acquire new capabilities from expert teachers. However, integrating knowledge from multi-teacher, multi-domain demonstrations into a single student remains challenging. We study supervised fine-tuning (SFT) in this setting, where students must acquire diverse capabilities while maintaining generalization beyond the training tasks. Our experiments reveal varying trade-offs between in-distribution learning and out-of-distribution generalization across SFT methods, motivating more explicit control over this balance. To this end, we propose soft-target fine-tuning (SoFT) to balance learning from teacher demonstrations with retaining the Base model's existing capabilities. SoFT sets a probability floor for each demonstrated token while making the smallest KL change to the Base distribution. The resulting objective couples learning from demonstrations with adaptively weighted regularization toward the Base model. We further use domain-specific gradient budgets to control this balance and calibrate a sequence-specific probability floor. Experiments on mixed-domain reasoning and agentic tasks show that SoFT achieves the best overall performance among the compared methods, with improvements in both in-distribution capability acquisition and out-of-distribution generalization.

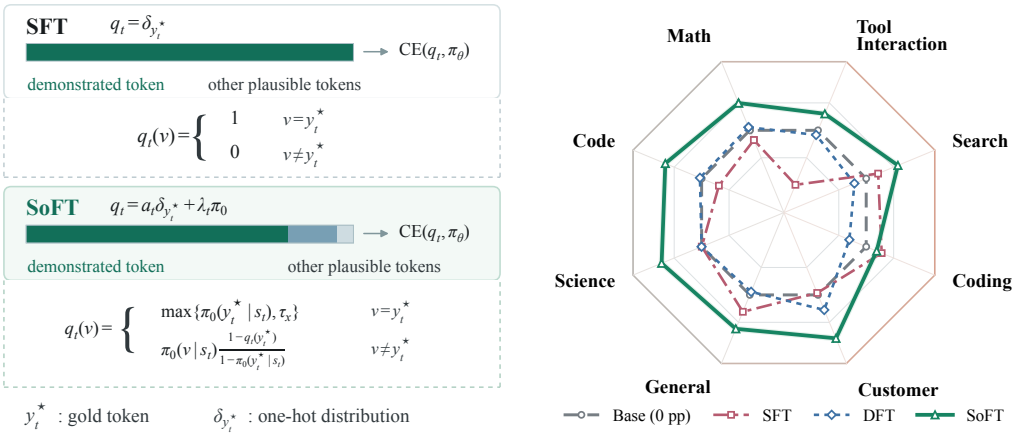

Figure 1: SoFT overview and mixed-data outcomes. Left: schematic targets; SFT uses a one-hot target, while SoFT retains Base-supported alternatives. Right: changes relative to Base across reasoning and agentic domains.

1 Introduction

Supervised fine-tuning (SFT) on expert demonstrations is a standard way to transfer new capabilities to a language model (Ouyang et al., 2022; Lambert et al., 2024). Demonstrations may contain intermediate reasoning (Muennighoff et al., 2025; DeepSeek-AI, 2025; Guha et al., 2025), tool decisions, or long multi-turn interactions (Raoof et al., 2026; Jung et al., 2026), and practical corpora often mix trajectories from several teachers and task domains (Wan et al., 2024; Guha et al., 2025). A single student must then acquire all of these capabilities without losing the useful behavior it learned during pretraining (Li et al., 2025; Wang et al., 2026b).

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.

Standard SFT is poorly suited to this goal. A teacher trajectory is only one of many valid continuations: another teacher, or the student itself, might use different wording, reasoning steps, tool calls, or stopping points (Kim et al., 2025; Raoof et al., 2026). Yet SFT trains toward a one-hot target on every observed token. Repeated updates therefore favor the demonstrated trajectory at the expense of other plausible continuations (Li et al., 2025; Chen et al., 2025), especially in long reasoning and agentic trajectories (Twist et al., 2026; Jung et al., 2026). SFT also ignores the student's starting point, pushing every demonstrated token toward certainty whether Base already predicts it well or finds it unlikely (Chen et al., 2025; Wang et al., 2026a). Under mixed data, these costs are uneven across domains: in Figure 1 (right), SFT and DFT improve some domains but fall below Base on others. Existing variants reweight or select training signals (Xia et al., 2024; Lin et al., 2024; Chen et al., 2025; Wu et al., 2026; Wang et al., 2026a) or anchor the student to its initialization (Li et al., 2018; Zhu et al., 2025; Wang et al., 2026b), but they typically tune learning and preservation separately (Section 2.3).

We propose soft-target fine-tuning (**SoFT**), which replaces the one-hot target with a soft target (Figure 1, left): the distribution closest to Base in KL that gives the demonstrated token at least a floor probability. This design has three advantages. First, it changes Base minimally: the target adds only the probability mass the floor requires and keeps Base's relative preferences among other tokens, so plausible continuations are not needlessly suppressed. Second, it couples learning and preservation: the target splits into complementary demonstration and Base-preservation weights, so supervision concentrates on tokens that Base poorly supports. Third, it is simple to control. A single gradient budget, defined as the fraction of SFT's initial gradient on demonstrated tokens that SoFT retains, sets the learning strength, and a per-trajectory floor adapts it to the student's initial probabilities. Training needs only offline demonstrations, with no teacher logits or online rollouts. Our contributions are threefold:

1. **Mixed-data SFT hides an acquisition–retention trade-off.** When one student learns from multiple teachers and domains, gains on some task families often come with sharp losses on others. At 7B, four of six fine-tuning baselines fall below Base in aggregate score, and at 8B, SFT gains in-distribution but loses out-of-distribution.
2. **SoFT turns the training target itself into the control for this trade-off.** We derive a closed-form, minimum-KL soft target that decomposes exactly into complementary learning and preservation weights, and calibrate it with a per-domain gradient budget. SoFT needs no teacher logits or online rollouts.
3. **SoFT gains more while drifting less.** SoFT achieves the best aggregate score in all three settings and improves 22 of 24 benchmark scores over Base, e.g., raising τ^2 OOD from 20.67 to 48.33. Its policy drift is only about one tenth of SFT's, and our analysis shows that stronger imitation of demonstrated tokens alone does not explain these gains.

2 Preliminary

2.1 Supervised Fine-Tuning

Let $\mathcal{D} = \{(x, y^\star)\}$ denote a corpus of expert demonstrations, where $y^\star$ is the complete reference response to query x. SFT minimizes the negative log-likelihood of the demonstrated response:

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = \mathbb{E}_{(x,y^\star)\sim\mathcal{D}}[-\log \pi_\theta(y^\star \mid x)]\,. \tag{1}$$

2.2 Applications of SFT

SFT remains a central stage in recent model training pipelines and transfers both reasoning and agentic capabilities from verified demonstrations. For reasoning, supervised rationales expose intermediate computations and decomposition strategies in addition to final answers (Ho et al., 2023; Magister et al., 2023; Hsieh et al., 2023; Shridhar et al., 2023; Yue et al., 2024). Recent recipes such as s1, OpenThoughts, and skill-aware distillation improve this transfer through data curation, scaling, and student-specific selection (Muennighoff et al., 2025; Guha et al., 2025; Zhang et al., 2026). DeepSeek-R1 and Qwen3 further demonstrate reasoning transfer from stronger models to smaller students (DeepSeek-AI, 2025; Yang et al., 2025). For agentic capabilities, trajectories supervise tool use, environment interaction, recovery, and task completion. Kimi K2.5, Kimi K3,

Qwen3-Coder-Next, DeepSeek V4.1, OpenThoughts-Agent, and ProCUA-SFT apply this approach to long-horizon and executable tasks (Kimi Team, 2026a;b; Cao et al., 2026; DeepSeek-AI, 2026; Raoof et al., 2026; Jung et al., 2026). Since trajectory SFT uses a fixed corpus and requires only teacher outputs, it supports black-box distillation and reuses each verified trajectory across many updates (Kim & Rush, 2016; Guha et al., 2025). These properties make SFT a scalable first stage for acquiring new model capabilities.

2.3 SFT Improvement

Many SFT improvements on a fixed demonstration corpus can be written in the general form

$$\mathcal{L}(\theta) = \mathbb{E}_{(x,y^\star)\sim\mathcal{D}}\left[\sum_t \left(-a_t \log \pi_\theta(y_t^\star \mid s_t) + \lambda_t \Omega_t(\pi_\theta(\cdot \mid s_t), \pi_0(\cdot \mid s_t))\right)\right], \tag{2}$$

where $s_t = (x, y^\star_{<t})$ is the teacher-forced state, a_t controls how strongly the demonstrated token is learned, and $\lambda_t\Omega_t$ controls the deviation of the updated policy π_θ from the Base policy π_0. Standard SFT is recovered by setting $a_t = 1$ and $\lambda_t = 0$.

The first line of work adjusts a_t to reallocate learning strength across examples or tokens. LESS, Rho-1, and PriFT-mass select training signals, while Confidence and DFT assign model-dependent weights (Xia et al., 2024; Lin et al., 2024; Wang et al., 2026a; Wu et al., 2026). These methods change how strongly each demonstration token is learned, but usually retain the one-hot target on $y_t^\star$. The second line adjusts λ_t or Ω_t to limit how far the student moves from the Base model. ASFT adds a KL anchor to DFT (Zhu et al., 2025), while Anchored Learning controls distributional drift through an intermediate target (Wang et al., 2026b). L2-SP is the parameter-space analogue, replacing the policy discrepancy above with an anchor between θ and its initialization θ_0 (Li et al., 2018). These methods preserve prior behavior through a regularizer whose strength is usually specified separately from the SFT signal.

SoFT couples these two lines. Its adaptive target is equivalent to jointly setting the demonstration coefficient a_t and the Base-preservation coefficient λ_t, so that a larger value of one leaves a smaller value for the other. This coupling controls acquisition and preservation through one calibrated target rather than two independently tuned objectives. Sections 3.2 and 3.4 define this target and its calibration.

3 Method

We introduce *soft-target fine-tuning* (SoFT), an offline objective that couples learning from demonstrations with preserving the Base policy. From the demonstrated sequence and the Base distribution alone, SoFT constructs one adaptive training target for each token, without online rollouts or teacher logits.

3.1 From a Global Budget to Token Targets

Figure 2 gives an overview, from left to right. A global gradient budget $R^\star$ (left) sets SoFT's initial gradient on demonstrated tokens relative to SFT. For each sequence, this budget is converted into a probability floor τ_x (middle). At each teacher-forced state $s_t = (x, y^\star_{<t})$, each position whose Base probability $\pi_0(y_t^\star \mid s_t)$ falls below τ_x is lifted to the floor, which determines a demonstration weight a_t; for example, with $\tau_x = 0.7$, a token with Base probability 0.1 receives $a_t = 2/3$. The weight then defines a token-level soft target (right) that, unlike the one-hot SFT target, keeps part of Base's probability on alternative tokens. Thus, $R^\star$ is selected globally, τ_x adapts to each sequence, and the target adapts to each token. Sections 3.2–3.4 formalize these three steps in reverse order: the target, its decomposition into weights, and the calibration of τ_x.

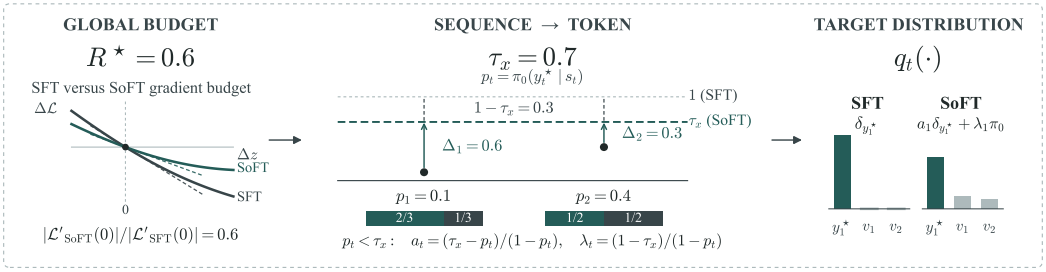

Figure 2: From a shared gradient budget (left), to a sequence-specific floor (middle), to token-level SoFT targets (right). Values are illustrative.

3.2 Soft Target Construction

At each teacher-forced state s_t, SoFT replaces the one-hot target with the distribution q_t closest to the Base policy π_0 that assigns at least probability τ_x to the demonstrated token $y_t^\star$ (we leave its dependence on τ_x implicit):

$$q_t = \arg\min_{q \in \Delta(\mathcal{V})} \mathrm{KL}(q \,\|\, \pi_0(\cdot \mid s_t)) \quad \text{s.t.} \quad q(y_t^\star) \geq \tau_x. \tag{3}$$

The KL term discourages unnecessary changes to the Base distribution, and the constraint enforces the probability floor. Solving the Lagrangian gives the closed form

$$q_t(y_t^\star) = \max\{\pi_0(y_t^\star \mid s_t), \tau_x\}, \qquad q_t(v) = \pi_0(v \mid s_t)\frac{1 - q_t(y_t^\star)}{1 - \pi_0(y_t^\star \mid s_t)}, \quad v \neq y_t^\star. \tag{4}$$

The solution adds only the probability mass required by the floor and rescales all other tokens proportionally, so Base's relative preferences among them are unchanged. If $\pi_0(y_t^\star \mid s_t) \geq \tau_x$, the constraint is inactive and $q_t = \pi_0(\cdot \mid s_t)$; as $\tau_x \to 1$, the target approaches the one-hot SFT target. Appendix A.1 gives the full derivation.

3.3 Soft Target Decomposition

We now decompose the soft target to connect it with the learning and preservation terms in Eq. 2. Letting $\delta_{y_t^\star}$ denote the one-hot distribution on $y_t^\star$, Eq. 4 can be written as

$$a_t = \frac{\max\{\tau_x - \pi_0(y_t^\star \mid s_t),\, 0\}}{1 - \pi_0(y_t^\star \mid s_t)}, \qquad \lambda_t = 1 - a_t, \qquad q_t = a_t\delta_{y_t^\star} + \lambda_t\pi_0(\cdot \mid s_t). \tag{5}$$

Here, a_t weights learning from the demonstrated token and λ_t weights preservation of the Base distribution, and the two are complementary by construction. Tokens that Base already predicts above the floor receive $a_t = 0$ and keep their Base target; all other tokens receive exactly the extra probability needed to reach τ_x.

Substituting Eq. 5 into the cross-entropy gives

$$\mathcal{L}_{\mathrm{SoFT}}(\theta) = \mathbb{E}_{(x,y^\star)\sim\mathcal{D}}\left[\sum_t \left(-a_t \log \pi_\theta(y_t^\star \mid s_t) + \lambda_t \mathrm{CE}(\pi_0(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t))\right)\right]. \tag{6}$$

Since this cross-entropy equals $\mathrm{KL}(\pi_0 \| \pi_\theta)$ up to a θ-independent constant, SoFT instantiates Eq. 2 with Ω_t as the forward KL, through a single soft target whose learning and preservation weights are coupled rather than tuned independently.

3.4 Adaptive Control with $R^\star$

The floor τ_x determines both weights in Eq. 5. Since the Base probabilities $\pi_0(y_t^\star \mid s_t)$ vary across demonstrations, a floor shared by all sequences would produce different effective learning strengths. We therefore solve for a sequence-specific floor that meets a shared gradient budget $R^\star$.

At the Base initialization, the gradient magnitude on the gold logit (the logit of $y_t^\star$) is $1-\pi_0(y_t^\star \mid s_t)$ for SFT and $a_t(1-\pi_0(y_t^\star \mid s_t))$ for SoFT. We choose τ_x such that

$$R_x(\tau_x) = \frac{\sum_t a_t(1-\pi_0(y_t^\star \mid s_t))}{\sum_t(1-\pi_0(y_t^\star \mid s_t))} = R^\star. \tag{7}$$

Each numerator term equals $q_t(y_t^\star) - \pi_0(y_t^\star \mid s_t)$, so $R^\star$ is the fraction of Base's gap to the one-hot target that the soft target closes (e.g., $R^\star = 0.3$ closes 30% of it); equivalently, it is the fraction of SFT's initial gold-logit gradient that SoFT retains. As $R^\star$ moves from zero to one, SoFT moves from preserving the Base distribution toward one-hot SFT. Since $R_x(\tau)$ is continuous and monotone in τ, τ_x is obtained by scalar bisection (Appendix A.4). Because q_t depends only on π_0, it stays fixed during training and can be precomputed; Appendix B.2.1 gives a memory-efficient top-K approximation.

4 Experimental Setup

We evaluate whether SoFT can learn from mixed training data while preserving generalization beyond the training tasks. We compare SoFT with Base, standard SFT (Ouyang et al., 2022), Confidence (Chen et al., 2025), Rho-1 (Lin et al., 2024), DFT (Wu et al., 2026), PriFT-mass (Wang et al., 2026a), and L2-SP (Li et al., 2018) on reasoning and agentic tasks. Within each setting, all fine-tuned methods share the initialization, data, and training budget, and hyperparameters are selected on held-out validation data.

4.1 Domain-Aware $R^\star$

For SoFT, we select one gradient budget $R_d^\star$ per domain using held-out validation data. Each sequence x then uses the budget of its domain $d(x)$:

$$R_x(\tau_x) = R^\star_{d(x)}, \tag{8}$$

so that $R_d^\star$ sets the domain-level balance and τ_x adapts it to each sequence. For General, Math, Science, and Code, Qwen2.5-1.5B-Instruct uses $R_d^\star = (0.15, 0.30, 0.45, 0.60)$ and Qwen2.5-7B uses $(0.25, 0.35, 0.45, 0.55)$. Qwen3-8B uses $(0.3, 0.8, 0.3, 0.3)$ for Customer, Coding, Search, and Tool Interaction.

4.2 Tasks and Evaluation

For reasoning, we fine-tune Qwen2.5-1.5B-Instruct and Qwen2.5-7B (Yang et al., 2024) on 36,000 teacher trajectories, equally divided among Math, Code, Science, and General, with 800 validation examples. For agentic tasks, we fine-tune Qwen3-8B (Yang et al., 2025) on 8,252 trajectories across Customer, Coding, Search, and Tool Interaction, with 824 validation examples. Teachers include DeepSeek-R1, GLM-5.1, Kimi-K2.5, QwQ-32B/STILL-2, Qwen3-235B-A22B, GPT-OSS-20B, DeepSeek V4 Flash, and Qwen3.6-27B (Appendix C). Within each capability domain, we use in-distribution (ID) for training-related evaluation and out-of-distribution (OOD) for cross-benchmark evaluation (Table 1).

Reasoning evaluations use OpenR1-Math (Lozhkov et al., 2025), APPS (Hendrycks et al., 2021), RiddleSense (Lin et al., 2021), GSM8K (Cobbe et al., 2021), ARC-Challenge (Clark et al., 2018), BoolQ (Clark et al., 2019), and MBPP with EvalPlus tests (Austin et al., 2021; Liu et al., 2023). Science MCQ is our filtered subset of Mixture-of-Thoughts (Hugging Face, 2025b), not a separate official benchmark. Agentic evaluations use τ^2 (Barres et al., 2025), LiveCodeBench (Jain et al., 2024), TMax (Ivison et al., 2026), BigCodeBench (Zhuo et al., 2024), BFCL V4 (Patil et al., 2025), and FRAMES (Krishna et al., 2024). For τ^2, training and the 60 ID tasks use the retail and airline domains, while the 300 OOD tasks come from telecom and banking knowledge. Search ID uses held-out Nemotron search prompts (NVIDIA, 2025).

All models use chain-of-thought reasoning, and we report empirical pass@4 over the temperature–seed pairs $(0, 42)$, $(0.2, 43)$, $(0.4, 44)$, and $(0.6, 45)$; a task counts as solved if any attempt passes the verifier. We aggregate the benchmark scores $s_b \in [0, 1]$ with the geometric mean $\mathrm{GM} = 100\big(\prod_{b=1}^{8} s_b\big)^{1/8}$, which weights the four ID and four OOD benchmarks equally (task details in Appendix D).

Table 1: Evaluation suites and case counts before repeated attempts. BigCodeBench counts evaluation cases.

Setting	Domain	ID (cases)	OOD (cases)
Reasoning	Math	OpenR1-Math (300)	GSM8K (1,319)
	Code	APPS (250)	MBPP+ (378)
	Science	Science MCQ (1,000)	ARC-Challenge (608)
	General	RiddleSense (1,018)	BoolQ (708)
Agentic	Customer	τ^2 (60)	τ^2 (300)
	Coding	LiveCodeBench (694)	BigCodeBench (2,280)
	Search	Search (500)	FRAMES (824)
	Tool Interaction	TMax (300)	BFCL V4 (5,106)

Table 2: Reasoning performance (%) under the fixed four-attempt schedule. Bold marks column bests per model size, including ties. Math: OpenR1-Math; Riddle: RiddleSense.

		In-distribution (ID)				**Out-of-distribution (OOD)**				
Model	Method	Math	APPS	Riddle	Science	GSM8K	ARC-C	BoolQ	MBPP+	**GM**
Qwen2.5-1.5B	Base	34.00	6.80	66.50	76.50	87.34	85.03	82.91	60.85	50.76
	SFT	37.00	**8.00**	70.73	78.30	87.11	88.49	82.77	63.76	53.46
	Confidence	33.33	6.00	71.81	79.00	85.06	**88.82**	83.05	62.70	50.84
	Rho-1	35.00	6.80	72.00	77.30	86.05	87.99	83.47	60.58	51.66
	DFT	31.00	6.40	68.66	73.40	83.55	86.35	81.21	60.58	49.40
	PriFT-mass	36.00	5.60	70.04	75.60	85.90	83.39	84.18	65.61	50.49
	L2-SP	35.33	6.80	70.53	79.20	86.28	**88.82**	82.91	65.87	52.32
	SoFT	**37.33**	**8.00**	**72.30**	**79.80**	**87.49**	86.18	**85.59**	**66.67**	**54.17**
Qwen2.5-7B	Base	52.00	12.80	82.61	82.70	91.74	92.11	92.80	68.78	62.91
	SFT	42.33	11.60	84.28	87.10	94.92	82.40	93.50	59.26	59.46
	Confidence	38.67	**13.20**	84.87	**88.30**	94.62	75.00	90.82	70.37	60.24
	Rho-1	24.33	10.40	76.72	85.60	89.39	79.77	83.47	30.16	48.34
	DFT	**58.00**	11.20	82.81	82.70	93.71	93.91	90.96	71.69	63.22
	PriFT-mass	55.33	12.80	83.30	83.60	94.69	95.23	92.23	74.34	64.64
	L2-SP	43.00	12.80	84.09	84.80	94.24	96.05	93.64	72.49	62.77
	SoFT	55.33	12.80	**85.56**	**88.30**	**95.38**	**97.37**	**94.35**	**75.66**	**65.88**

5 Experimental Results

Tables 2 and 3 report the results. SoFT achieves the highest GM in all three settings, while individual benchmarks reveal the distinct strengths of other methods.

5.1 Main Results on Reasoning Tasks

Challenges of mixed-data training. Fine-tuning gains are uneven across reasoning tasks (Table 2). At 7B, Confidence leads on APPS and ties for the best Science score, yet its Math score falls from 52.00 to 38.67 and ARC-C from 92.11 to 75.00, leaving its GM (60.24) below that of Base (62.91). SFT likewise improves Science but loses more than nine points on both Math and MBPP+. Overall, four baselines at 7B and two at 1.5B finish below Base in GM: training on the mixture can strengthen one part of it while weakening another.

SoFT under mixed-data training. SoFT improves nearly every reasoning benchmark. It raises all eight scores over Base at 1.5B and seven at 7B, matching Base on the remaining task, and its GMs of 54.17 and 65.88 are the highest at both scales. At 7B, Science rises from 82.70 to 88.30 and MBPP+ from 68.78 to 75.66. DFT remains strongest on 7B Math and Confidence on 7B APPS, but SoFT's broader gains yield the highest aggregate score.

Table 3: Agentic performance (%) under the fixed four-attempt schedule. Bold marks column bests, including ties. LCB: LiveCodeBench; BCB: BigCodeBench.

		In-distribution (ID)				Out-of-distribution (OOD)				
Model	Method	τ^2	LCB	Search	TMax	τ^2	BCB	BFCL V4	FRAMES	**GM**
Qwen3-8B	Base	48.33	39.05	23.80	7.33	20.67	53.51	40.20	46.12	30.11
	SFT	35.00	37.32	26.20	**16.00**	32.00	**58.20**	21.10	46.00	31.58
	Confidence	40.00	35.59	17.60	5.00	22.00	56.32	34.15	48.67	26.70
	Rho-1	58.33	**40.20**	22.80	9.00	29.33	53.51	40.60	45.15	32.93
	DFT	56.67	37.32	23.00	5.33	27.67	52.02	41.32	44.66	30.18
	PriFT-mass	56.67	39.48	20.40	7.67	26.00	52.89	40.02	43.93	30.97
	L2-SP	26.67	37.90	18.20	7.67	29.00	55.57	33.92	48.54	27.97
	SoFT	**65.00**	38.04	**27.20**	9.33	**48.33**	56.52	**41.36**	**48.79**	**36.93**

5.2 Results on Agentic Tasks

Challenges of mixed-data training. Agentic tasks show a similar imbalance (Table 3). SFT leads on TMax and BigCodeBench, yet its τ^2 ID score falls from 48.33 to 35.00 and its BFCL V4 score from 40.20 to 21.10. Confidence and L2-SP also finish below Base in GM, so coding and terminal gains can come at the cost of customer interaction and tool calling.

SoFT under mixed-data training. SoFT improves seven of eight agentic scores over Base and achieves the highest GM, 36.93. Its largest gains come from τ^2: ID rises from 48.33 to 65.00 and OOD from 20.67 to 48.33. The only regression is a slight dip on LiveCodeBench, from 39.05 to 38.04, while SFT still leads on TMax and BigCodeBench.

5.3 Shared Patterns across Task Families

Across task families, the key difference is not whether a method achieves a high score on one benchmark, but whether it does so without large losses elsewhere. For example, 8B SFT raises the four-benchmark ID geometric mean from 23.95 to 27.20 but lowers the OOD mean from 37.84 to 36.67.

SoFT instead raises both means in every setting: to 28.14 and 48.45 at 8B, from 32.93 to 36.23 and 78.24 to 80.99 at 1.5B, and from 46.18 to 48.10 and 85.70 to 90.23 at 7B. Overall, it improves 22 of 24 scores over Base, matches one, and falls slightly below on one.

6 Exploratory Analysis

We examine three consequences of SoFT's coupled objective (Eq. 6). **RQ1** relates benchmark gains to policy drift from Base. **RQ2** measures how much probability the trained model assigns to demonstrated tokens. **RQ3** examines how $R^\star$ distributes target changes across domains.

6.1 RQ1: Performance Gain versus Policy Drift

SoFT gains more with less policy drift. Figure 3 compares the GM gain of each trained model with its policy drift from Base. We measure drift as $\mathrm{KL}(\pi_0 \| \pi_\theta)$ on held-out teacher-forced prefixes and divide by SFT's KL within the same setting, so SFT lies at one on the horizontal axis. SoFT occupies the favorable upper-left region at all three scales. Its relative drift is about one tenth of SFT's, and its GM gain is the largest in each setting, peaking at 8B.

Output similarity and task success capture different forms of preservation. On reasoning tasks, SoFT's output-token frequencies stay closer to Base than SFT's at both scales (mean Jensen–Shannon divergence: 0.044 versus 0.090 at 1.5B; 0.065 versus 0.169 at 7B). SoFT's response length is also closer to Base than SFT's on 15 of 16 benchmarks (Figure 4). L2-SP offers a useful counterexample at 7B: its token frequencies are closer to Base, yet it retains fewer Base successes. Lexical

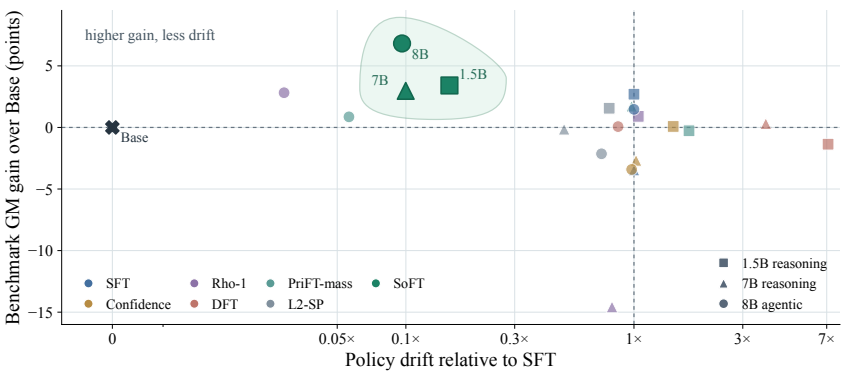

Figure 3: Benchmark GM gain versus policy drift relative to SFT. The pale rounded region highlights the three SoFT points. Colors denote methods, shapes denote settings, and the horizontal axis uses a symmetric log scale near zero.

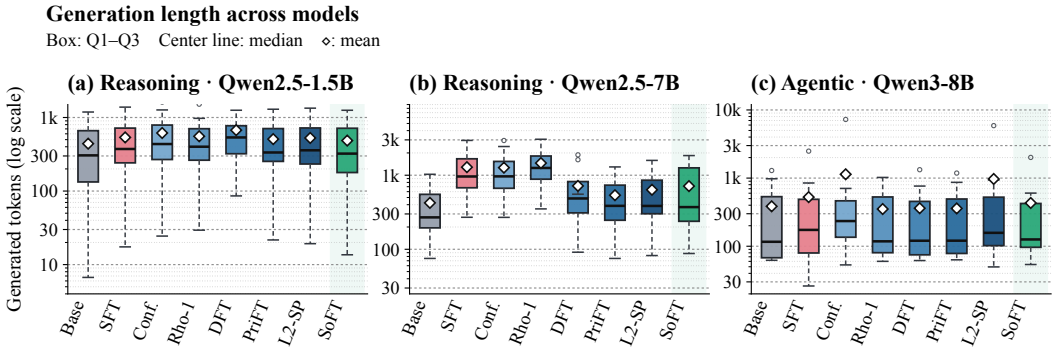

Figure 4: First-attempt output lengths for all eight methods across eight benchmark means per setting. Boxes: Q1–Q3; center lines: medians; diamonds: means. Agentic lengths cover scorer-facing outputs, not total rollout cost.

similarity and task success thus describe different aspects of preservation. Appendix F reports the absolute KL values and a paired retention analysis.

6.2 RQ2: Realized Demonstration Learning

More demonstration uptake does not guarantee better performance. We measure how training changes the probability of demonstrated tokens at shared teacher-forced states. We define $A_\theta = \sum_t[\pi_\theta(y_t^\star \mid s_t) - \pi_0(y_t^\star \mid s_t)] / \sum_t[1 - \pi_0(y_t^\star \mid s_t)]$. It measures the share of Base's remaining gold-token probability (Base headroom) acquired by the trained model; on SoFT's constructed target, it equals $R^\star_{d(x)}$. Figure 5 computes A_θ on the first 128 tokens of 32 validation trajectories per reasoning domain. At 1.5B, SoFT has the smallest A_θ (7.2%) and the highest GM (54.17), while DFT has the largest A_θ (41.2%) and a lower GM (49.40). At 7B, PriFT-mass, Confidence, and Rho-1 have acquisition ratios near 36%, while their GMs span 48.34 to 64.64. A larger A_θ thus does not imply better task performance, and SoFT's high GM with a small A_θ is consistent with preserving Base-supported alternatives. Appendix I gives per-domain results.

6.3 RQ3: Effect of the Gradient Budget

A shared budget produces comparable relative adjustments. We vary $R^\star$ and measure $\mathrm{KL}(q_t \| \pi_0)$ for the resulting targets. In Figure 6, Science requires a larger absolute change than Math at $R^\star = 0.5$ on 1.5B (0.739 versus 0.363 nats per token). The lower panels normalize each trajectory's KL by its value at the one-hot endpoint. At the same budget, the domain averages cluster within 41.6–42.6% at 1.5B and 48.4–49.9% across the four agentic groups at 8B. Domains with different absolute target changes thus use similar fractions of their available change, making $R^\star$ an interpretable control across the mixture.

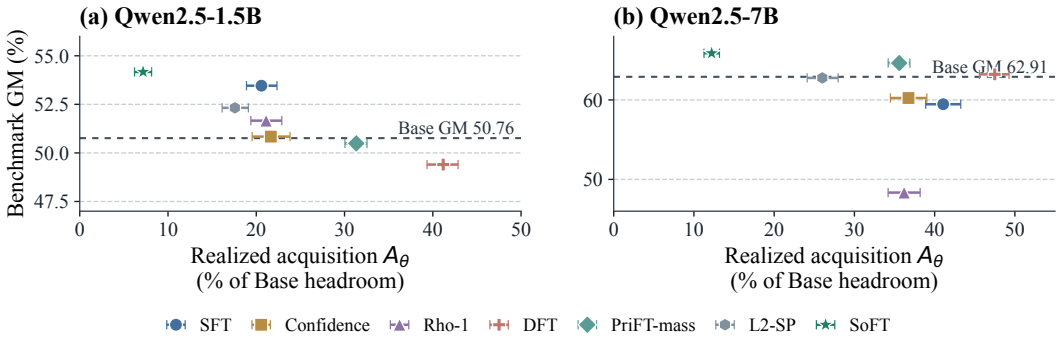

Figure 5: Realized acquisition A_θ versus benchmark GM for seven methods at each reasoning scale. Horizontal bars show exploratory resampling ranges; dashed lines mark Base GM. The validation subset was used during hyperparameter selection.

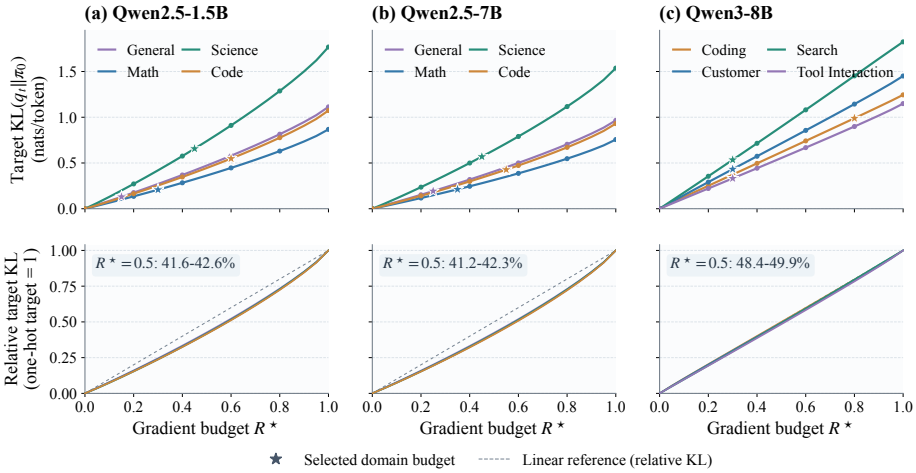

Figure 6: Target KL from Base as the gradient budget varies. Top: absolute KL of the constructed target; bottom: KL relative to each trajectory's one-hot endpoint. Stars mark selected budgets.

The same budget activates different fractions of tokens. At $R^\star = 0.3$, positive demonstration weights apply to 16.9% of Search tokens and 19.8% of Customer tokens. Even within Tool Interaction, the Terminal and Tool Calling sources differ more sharply (9.3% versus 25.5%). Tokens that Base already predicts above the sequence floor keep their Base target, while tokens with lower Base probability receive more weight. As the budget rises, the solved floor activates additional positions rather than raising every weight equally, letting a domain budget respond to its Base confidence profile (Appendices E and G).

Selected budgets express domain priorities. The coding domain receives the largest budget in every setting (stars in Figure 6); validation sets each domain's learning strength, and the sequence-specific floor adapts it to Base confidence.

7 Conclusion

We studied multi-teacher, multi-domain SFT, where a student must learn diverse behaviors without losing generalization. SoFT uses calibrated soft targets to couple learning from demonstrations with preservation of the Base distribution. It has the best aggregate score across reasoning and agentic tasks in all three settings, although other methods lead on some benchmarks. The analysis also shows that stronger imitation of demonstrated tokens alone does not track the best aggregate task performance. These results suggest that controlling learning and preservation together can improve mixed-data fine-tuning without teacher logits or online rollouts. Future work will test whether this helps subsequent reinforcement learning. The broader lesson is to calibrate new supervision against existing knowledge.

AI USE STATEMENT

Language models were used as objects of study and to generate teacher trajectories for the experiments described in this paper. Beyond these experimental uses, generative AI tools were used only to assist with implementing parts of the experimental code. They were not used to draft, edit, or revise the manuscript, nor to generate the text of tables or figures. The authors reviewed all experimental code, including AI-assisted implementations, to check its correctness and consistency with the methods described in the paper. All analyses, interpretations of the experimental results, and scientific claims were developed and assessed by the authors. The authors take full responsibility for the experimental methodology, code, results, claims, and final manuscript.

ETHICS STATEMENT

This work studies how language models learn from and retain capabilities after distillation. The experiments use published datasets and benchmark environments, together with generated teacher trajectories; they do not involve direct interaction with human participants. Model outputs and published trajectories may nonetheless reflect errors, biases, or sensitive content present in their sources. We report benchmark-specific limitations and avoid treating aggregate scores as evidence that every capability improves. The authors are responsible for respecting the licenses and usage conditions of the datasets and models used in this work.

REPRODUCIBILITY STATEMENT

The main text specifies the training objectives, model families, hyperparameter-selection procedure, and evaluation protocols. Appendix C documents the training-data sources, teacher provenance, split sizes, token statistics, and examples. We use fixed data splits and seeds, shared evaluation tasks within each model-size comparison, and fixed scoring denominators. We distinguish the four-attempt scores, versioned answer-extraction procedures, and scale-specific finalization settings. Upon acceptance, we will publicly release the training and evaluation code, configurations, data manifests, and processed datasets that we are permitted to redistribute. For third-party data that cannot be redistributed, we will provide source links and reconstruction scripts.

REFERENCES

Wasi Uddin Ahmad, Sean Narenthiran, Somshubra Majumdar, Aleksander Ficek, Siddhartha Jain, Jocelyn Huang, Vahid Noroozi, and Boris Ginsburg. OpenCodeReasoning: Advancing data distillation for competitive coding. *arXiv preprint arXiv:2504.01943*, 2025. URL https://arxiv.org/abs/2504.01943.

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

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-Bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025. URL https://arxiv.org/abs/2506.07982.

Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, et al. Llama-Nemotron: Efficient reasoning models. *arXiv preprint arXiv:2505.00949*, 2025. URL https://arxiv.org/abs/2505.00949.

Ruisheng Cao, Mouxiang Chen, Jiawei Chen, Zeyu Cui, Yunlong Feng, Binyuan Hui, Yuheng Jing, Kaixin Li, Mingze Li, Junyang Lin, Zeyao Ma, Kashun Shum, Xuwu Wang, Jinxi Wei, Jiaxi Yang, Jiajun Zhang, Lei Zhang, Zongmeng Zhang, Wenting Zhao, and Fan Zhou. Qwen3-Coder-Next technical report. *arXiv preprint arXiv:2603.00729*, 2026. URL https://arxiv.org/abs/2603.00729.

Feng Chen, Allan Raventos, Nan Cheng, Surya Ganguli, and Shaul Druckmann. Rethinking fine-tuning when scaling test-time compute: Limiting confidence improves mathematical reasoning. *arXiv preprint arXiv:2502.07154*, 2025. URL https://arxiv.org/abs/2502.07154.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019. URL https://arxiv.org/abs/1905.10044.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

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. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression. *Technical report*, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4.1-Flash/blob/main/DeepSeek_V41_Tech_Report.pdf.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. OpenThoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025. URL https://arxiv.org/abs/2506.04178.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. Measuring coding challenge competence with APPS. *arXiv preprint arXiv:2105.09938*, 2021. URL https://arxiv.org/abs/2105.09938.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 14852–14882, 2023.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017, 2023.

Hugging Face. Open R1: A fully open reproduction of DeepSeek-R1, 2025a. URL https://github.com/huggingface/open-r1.

Hugging Face. Mixture-of-Thoughts. Hugging Face dataset, 2025b. URL https://huggingface.co/datasets/open-r1/Mixture-of-Thoughts.

iAmBoosted. GPT-OSS-20B reasoning traces. Hugging Face dataset, 2026. URL https://huggingface.co/datasets/iAmBoosted/gpt-oss-20b-reasoning-traces.

ianncity. KIMI-K2.5-1000000x. Hugging Face dataset, 2026. URL https://huggingface.co/datasets/ianncity/KIMI-K2.5-1000000x.

Hamish Ivison, Junjie Oscar Yin, Rulin Shao, Teng Xiao, Nathan Lambert, and Hannaneh Hajishirzi. Tmax: A simple recipe for terminal agents. *arXiv preprint arXiv:2606.23321*, 2026. URL https://arxiv.org/abs/2606.23321.

Jackrong. GLM-5.1-Reasoning-1M-Cleaned. Hugging Face dataset, 2026a. URL https://huggingface.co/datasets/Jackrong/GLM-5.1-Reasoning-1M-Cleaned.

Jackrong. Kimi-K2.5-Reasoning-1M-Cleaned. Hugging Face dataset, 2026b. URL https://huggingface.co/datasets/Jackrong/Kimi-K2.5-Reasoning-1M-Cleaned.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024. URL https://arxiv.org/abs/2403.07974.

Jaehun Jung, Ximing Lu, Brandon Cui, Muhammad Khalifa, Shaokun Zhang, Hao Zhang, Jin Xu, Amala Sanjay Deshmukh, Karan Sapra, Andrew Tao, Yejin Choi, Jan Kautz, Mingjie Liu, and Yi Dong. ProCUA-SFT technical report. *arXiv preprint arXiv:2606.17321*, 2026. URL https://arxiv.org/abs/2606.17321.

Kassadin88. GLM-5.1-1000000x: One million reasoning traces distilled from GLM-5.1. Hugging Face dataset, 2026. URL https://huggingface.co/datasets/Kassadin88/GLM-5.1-1000000x. Original citation name retained; repository now redirects to clzoro.

Jaehoon Kim, Kwangwook Seo, and Dongha Lee. In their own words: Reasoning traces tailored for small models make them better reasoners. *arXiv preprint arXiv:2509.22230*, 2025. URL https://arxiv.org/abs/2509.22230.

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

Kimi Team. Kimi K2.5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026a. URL https://arxiv.org/abs/2602.02276.

Kimi Team. Kimi K3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026b. URL https://arxiv.org/abs/2607.24653.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. *arXiv preprint arXiv:2409.12941*, 2024. URL https://arxiv.org/abs/2409.12941.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. Preserving diversity in supervised fine-tuning of large language models. In *International Conference on Learning Representations*, 2025.

Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. RiddleSense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge. *arXiv preprint arXiv:2101.00376*, 2021. URL https://arxiv.org/abs/2101.00376.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. RHO-1: Not all tokens are what you need. In *Advances in Neural Information Processing Systems*, 2024.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. *arXiv preprint arXiv:2305.01210*, 2023. URL https://arxiv.org/abs/2305.01210.

Anton Lozhkov, Hynek Kydlíček, Loubna Ben Allal, Guilherme Penedo, Edward Beeching, Quentin Gallouédec, Nathan Habib, Lewis Tunstall, and Leandro von Werra. OpenR1-Math-220k. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/open-r1/OpenR1-Math-220k.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 1773–1781, 2023.

Marin Community. OpenThoughts4 11K Math - Qwen3-235B-A22B Agreed Answers. Hugging Face dataset, 2026. URL https://huggingface.co/datasets/marin-community/open-thoughts-4-11k-math-qwen3-235b-a22b-agreed-answers.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025. URL https://arxiv.org/abs/2501.19393.

NovaSky. Sky-T1_data_17k. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/NovaSky-AI/Sky-T1_data_17k.

NVIDIA. Nemotron-SFT-Agentic-v2. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/nvidia/Nemotron-SFT-Agentic-v2. Creation year follows the dataset card; the Hugging Face repository was created in March 2026.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025. URL https://arxiv.org/abs/2508.10925.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 48371–48392. PMLR, 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

Guilherme Penedo, Anton Lozhkov, Hynek Kydlíček, Loubna Ben Allal, Edward Beeching, Agustín Piqueres Lajarín, Quentin Gallouédec, Nathan Habib, Lewis Tunstall, and Leandro von Werra. CodeForces CoTs. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/open-r1/codeforces-cots.

Negin Raoof et al. Data recipes for agentic models. *arXiv preprint arXiv:2606.24855*, 2026.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. Distilling reasoning capabilities into smaller language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 7059–7073, 2023.

Lukas Twist, Helen Yannakoudakis, and Jie M. Zhang. Reasoning-trace collapse: Evaluating the loss of explicit reasoning during fine-tuning. *arXiv preprint arXiv:2605.21127*, 2026. URL https://arxiv.org/abs/2605.21127.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. Knowledge fusion of large language models. In *International Conference on Learning Representations*, 2024.

Ke Wang, Shuangqi Li, Mathieu Salzmann, and Pascal Frossard. PriFT: Prior-support guided supervised fine-tuning. *arXiv preprint arXiv:2606.09396*, 2026a.

Xinyu Wang, Changzhi Sun, Yuanbin Wu, and Xiaoling Wang. Stabilizing LLM supervised fine-tuning via explicit distributional control. *arXiv preprint arXiv:2605.04468*, 2026b. URL https://arxiv.org/abs/2605.04468.

Yongliang Wu, Yizhou Zhou, Ziheng Zhou, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, and Xu Yang. On the generalization of SFT: A reinforcement learning perspective with reward rectification. In *International Conference on Learning Representations*, 2026.

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*, pp. 54104–54132, 2024.

An Yang et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MAmmoTH: Building math generalist models through hybrid instruction tuning. In *International Conference on Learning Representations*, 2024.

Lechen Zhang, Yunxiang Zhang, Wei Hu, and Lu Wang. Skill-aware data selection and fine-tuning for data-efficient reasoning distillation. *arXiv preprint arXiv:2601.10109*, 2026. URL https://arxiv.org/abs/2601.10109.

He Zhu, Junyou Su, Peng Lai, Ren Ma, Wenjia Zhang, Linyi Yang, and Guanhua Chen. Anchored supervised fine-tuning. *arXiv preprint arXiv:2509.23753*, 2025. URL https://arxiv.org/abs/2509.23753.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen Gong, Thong Hoang, Armel Randy Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, Prateek Yadav, Naman Jain, Alex Gu, Zhoujun Cheng, Jiawei Liu, Qian Liu, Zijian Wang, Binyuan Hui, Niklas Muennighoff, David Lo, Daniel Fried, Xiaoning Du, Harm de Vries, and Leandro Von Werra. BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions. *arXiv preprint arXiv:2406.15877*, 2024. URL https://arxiv.org/abs/2406.15877.

A THEORETICAL ANALYSIS OF SOFT

This section derives the SoFT target as a constrained KL projection, connects it to token-weighted SFT and Base-policy regularization, distinguishes it from DFT, and interprets the gradient budget $R^\star$.

Fix the teacher-forced state $s_t = (x, y^\star_{<t})$ used in Section 3.2, and let $y^\star_t$ be its demonstrated next token. We retain the main-text notation q_t, a_t, and $\lambda_t = 1 - a_t$ throughout.

A.1 CONSTRAINED KL PROJECTION

SoFT sets a probability floor τ_x for the demonstrated token $y^\star_t$. Among all targets that satisfy this floor, it selects the one closest to Base:

$$q_t = \arg\min_{q \in \Delta(\mathcal{V})} \mathrm{KL}(q \,\|\, \pi_0(\cdot \mid s_t)) \quad \text{s.t.} \quad q(y^\star_t) \geq \tau_x. \tag{9}$$

When $\pi_0(y^\star_t \mid s_t) \geq \tau_x$, Base already satisfies the constraint and gives $q_t = \pi_0(\cdot \mid s_t)$. When $\pi_0(y^\star_t \mid s_t) < \tau_x$, the target assigns τ_x to $y^\star_t$. The Lagrangian determines the remaining mass:

$$\mathcal{J}(q, \eta_t) = \sum_{v \neq y^\star_t} q(v) \log \frac{q(v)}{\pi_0(v \mid s_t)} + \eta_t \left(\sum_{v \neq y^\star_t} q(v) - (1 - \tau_x) \right). \tag{10}$$

Stationarity gives

$$\log \frac{q(v)}{\pi_0(v \mid s_t)} + 1 + \eta_t = 0, \qquad v \neq y^\star_t, \tag{11}$$

so all non-demonstrated probabilities must be rescaled by the same factor. Normalization then yields

$$\boxed{\begin{aligned} q_t(y^\star_t) &= \max\{\pi_0(y^\star_t \mid s_t), \tau_x\}, \\ q_t(v) &= \pi_0(v \mid s_t) \frac{1 - q_t(y^\star_t)}{1 - \pi_0(y^\star_t \mid s_t)}, \qquad v \neq y^\star_t. \end{aligned}} \tag{12}$$

For any $u, v \neq y^\star_t$, the solution preserves $q_t(u)/q_t(v) = \pi_0(u \mid s_t)/\pi_0(v \mid s_t)$. Thus, SoFT makes the minimum KL change needed to support the demonstrated token while keeping Base's relative preferences among alternatives.

A.2 SOFT TARGET DECOMPOSITION

Using the main-text coefficients,

$$\begin{aligned} a_t &= \begin{cases} \dfrac{\tau_x - \pi_0(y^\star_t \mid s_t)}{1 - \pi_0(y^\star_t \mid s_t)}, & \pi_0(y^\star_t \mid s_t) < \tau_x, \\ 0, & \pi_0(y^\star_t \mid s_t) \geq \tau_x, \end{cases} \qquad \lambda_t = 1 - a_t, \\ q_t &= a_t \delta_{y^\star_t} + \lambda_t \pi_0(\cdot \mid s_t). \end{aligned} \tag{13}$$

The mixture assigns $\max\{\pi_0(y^\star_t \mid s_t), \tau_x\}$ to $y^\star_t$ and rescales the remaining Base mass proportionally, exactly recovering Eq. 12.

Let $\mathcal{L}_{\mathrm{SFT},t} = -\log \pi_\theta(y^\star_t \mid s_t)$. Using the linearity of cross-entropy in its target,

$$\mathcal{L}_{\mathrm{SoFT},t} = \mathrm{CE}(q_t, \pi_\theta(\cdot \mid s_t)) \tag{14}$$

$$= a_t \mathrm{CE}\big(\delta_{y^\star_t}, \pi_\theta(\cdot \mid s_t)\big) + \lambda_t \mathrm{CE}(\pi_0(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)) \tag{15}$$

$$= a_t \mathcal{L}_{\mathrm{SFT},t} + \lambda_t \mathrm{KL}(\pi_0(\cdot \mid s_t) \,\|\, \pi_\theta(\cdot \mid s_t)) + \lambda_t H(\pi_0(\cdot \mid s_t)). \tag{16}$$

The Base entropy is independent of θ. Therefore,

$$\boxed{\mathcal{L}_{\mathrm{SoFT},t} = a_t \mathcal{L}_{\mathrm{SFT},t} + \lambda_t \mathrm{KL}(\pi_0(\cdot \mid s_t) \,\|\, \pi_\theta(\cdot \mid s_t)) + \text{constant}.} \tag{17}$$

SoFT combines demonstrated-token learning with Base-policy preservation at each token. The probability floor determines the strength of both terms through $a_t + \lambda_t = 1$.

A.3 Relationship to DFT

Let z_t denote the student logits at state s_t. From Eq. 13, the SoFT gradient is

$$\nabla_{z_t}\mathcal{L}_{\mathrm{SoFT},t} = a_t\big(\pi_\theta(\cdot \mid s_t) - \delta_{y_t^\star}\big) + \lambda_t(\pi_\theta(\cdot \mid s_t) - \pi_0(\cdot \mid s_t))\,. \tag{18}$$

At initialization, $\pi_\theta(\cdot \mid s_t) = \pi_0(\cdot \mid s_t)$, so

$$\nabla_{z_t}\mathcal{L}_{\mathrm{SoFT},t}|_{\theta=\theta_0} = a_t\big(\pi_0(\cdot \mid s_t) - \delta_{y_t^\star}\big)\,. \tag{19}$$

Thus, the initial SoFT update is identical to token-weighted SFT with detached weight a_t.

DFT rescales the one-hot SFT gradient by the current student probability of the demonstrated token (Wu et al., 2026):

$$\begin{aligned}\mathcal{L}_{\mathrm{DFT},t} &= -\,\mathrm{sg}[\pi_\theta(y_t^\star \mid s_t)]\log \pi_\theta(y_t^\star \mid s_t),\\ \nabla_{z_t}\mathcal{L}_{\mathrm{DFT},t} &= \mathrm{sg}[\pi_\theta(y_t^\star \mid s_t)]\,\big(\pi_\theta(\cdot \mid s_t) - \delta_{y_t^\star}\big)\,.\end{aligned} \tag{20}$$

Here, sg denotes the stop-gradient operator. At initialization, DFT uses the weight $\pi_0(y_t^\star \mid s_t)$, whereas SoFT uses a_t.

For a closer comparison, consider weighted SFT with the fixed weight a_t, $\mathcal{L}_{\mathrm{matched},t} = -a_t \log \pi_\theta(y_t^\star \mid s_t)$, which has the same initial gradient as SoFT. As the student moves away from Base, the two gradients differ by

$$\nabla_{z_t}\mathcal{L}_{\mathrm{SoFT},t} - \nabla_{z_t}\mathcal{L}_{\mathrm{matched},t} = \lambda_t(\pi_\theta(\cdot \mid s_t) - \pi_0(\cdot \mid s_t))\,. \tag{21}$$

DFT and weighted SFT only rescale updates toward the one-hot target $\delta_{y_t^\star}$. SoFT additionally adds a restoring gradient toward Base and is minimized at q_t, so it changes both the strength and the destination of the update.

A.4 Interpretation of $R^\star$

At initialization, the gold-logit gradient magnitudes of SFT and SoFT are

$$\begin{aligned}\left|\frac{\partial \mathcal{L}_{\mathrm{SFT},t}}{\partial z_{t,y_t^\star}}\right| &= 1 - \pi_0(y_t^\star \mid s_t),\\ \left|\frac{\partial \mathcal{L}_{\mathrm{SoFT},t}}{\partial z_{t,y_t^\star}}\right| &= a_t(1 - \pi_0(y_t^\star \mid s_t)) = \max\{\tau_x - \pi_0(y_t^\star \mid s_t), 0\}.\end{aligned} \tag{22}$$

SoFT chooses the sequence-level floor τ_x such that

$$\boxed{R^\star = \frac{\sum_t a_t(1 - \pi_0(y_t^\star \mid s_t))}{\sum_t (1 - \pi_0(y_t^\star \mid s_t))} = \frac{\sum_t \max\{\tau_x - \pi_0(y_t^\star \mid s_t), 0\}}{\sum_t (1 - \pi_0(y_t^\star \mid s_t))}.} \tag{23}$$

Hence, $R^\star$ is the fraction of the initial summed SFT gold-logit update that SoFT retains. Since the non-gold gradient components are nonnegative and sum to the magnitude of the gold component, the same ratio also holds for the summed logit-gradient ℓ_1 norms.

The right-hand side of Eq. 23 is continuous, piecewise linear, and monotone in τ_x. Scalar bisection therefore finds the floor efficiently. Note that the ratio describes the initial logit gradients; the ratio of full parameter gradients can change during training.

B Minimal PyTorch Implementations

Both snippets assume a standard causal-LM batch: `labels` is -100 at prompt and padding positions, and shifting by one position produces the teacher-forced next-token pairs.

B.1 STANDARD SFT

Standard SFT gathers the log-probability of the demonstrated token, which is equivalent to cross-entropy with the one-hot target $q_t^{\mathrm{SFT}} = \delta_{y_t^\star}$.

```
import torch.nn.functional as F

def sft_loss(student, ids, mask, labels):
    logits = student(ids, attention_mask=mask).logits[:, :-1]
    gold = labels[:, 1:]
    keep = gold.ne(-100)
    return F.cross_entropy(logits[keep], gold[keep])
```

B.2 FIXED-$R^\star$ SOFT

SoFT follows Eqs. 4–7, using a frozen, evaluation-mode copy of the initial student as the reference model.

B.2.1 PRACTICAL APPROXIMATION

A full-vocabulary target costs $O(BT|\mathcal{V}|)$ memory. We retain the Base top-K tokens and the demonstrated token at each supervised position, then merge the remaining vocabulary into one tail bucket. With $\mathcal{S}_t = \{y_t^\star\} \cup \mathrm{TopK}(\pi_0(\cdot \mid s_t))$, the loss becomes

$$\widetilde{\mathcal{L}}_t = -\sum_{v \in \mathcal{S}_t} q_t(v) \log \pi_\theta(v \mid s_t) - q_t(\mathrm{tail}) \log \pi_\theta(\mathrm{tail} \mid s_t). \tag{24}$$

We use $K = 32$ and include $y_t^\star$ once when it appears in the top-K. The approximate target keeps the exact demonstrated-token probability and the total tail mass, representing the remaining tokens as one bucket, which reduces target storage to $O(BTK)$. Caching Base probabilities avoids repeated reference forward passes, and processing student positions in chunks limits activation memory. Inference uses the student alone.

The following dense reference implementation shows the exact fixed-$R^\star$ objective before the top-K approximation.

```
import torch
import torch.nn.functional as F

def soft_loss(student, reference, ids, mask, labels, rstar):
    logits = student(ids, attention_mask=mask).logits[:, :-1]
    gold = labels[:, 1:]
    keep = gold.ne(-100)
    gold = gold.masked_fill(~keep, 0)

    with torch.no_grad():
        ref_logits = reference(
            ids, attention_mask=mask
        ).logits[:, :-1]
        p0 = F.softmax(ref_logits, dim=-1)
        p0_gold = p0.gather(-1, gold[..., None]).squeeze(-1)

        # Solve R_x(tau_x) = R* for every sequence.
        weight = keep.to(p0.dtype)
        denom = ((1 - p0_gold) * weight).sum(-1).clamp_min(1e-12)
        lo, hi = torch.zeros_like(denom), torch.ones_like(denom)
        for _ in range(28):
            tau = (lo + hi) / 2
            numer = (((tau[:, None] - p0_gold).clamp_min(0))
```

```
                    * weight).sum(-1)
            right = numer / denom < rstar
            lo = torch.where(right, tau, lo)
            hi = torch.where(right, hi, tau)
        tau = ((lo + hi) / 2)[:, None]

        # Minimum-change target q_tau.
        q_gold = torch.maximum(p0_gold, tau)
        scale = (1 - q_gold) / (1 - p0_gold).clamp_min(1e-12)
        q = p0 * scale[..., None]
        q.scatter_(-1, gold[..., None], q_gold[..., None])

    logp = F.log_softmax(logits, dim=-1)
    token_loss = -(q * logp).sum(-1)
    return token_loss[keep].mean()
```

The Base probabilities define the target, the bisection loop converts the fixed budget $R^\star$ into one floor τ_x per sequence, and the last lines compute $\mathrm{CE}(q_t, \pi_\theta)$.

C Training Data Details

Training uses one epoch and seed 42, with maximum sequence lengths of 8,192 for reasoning and 32,768 for agentic tasks. Methods within each setting share the data order. The selected SoFT learning rates are 10^{-5}, 5×10^{-6}, and 10^{-6} for Qwen2.5-1.5B-Instruct, Qwen2.5-7B, and Qwen3-8B, respectively.

Counts measure training trajectories; several trajectories may share a task. Token statistics exclude validation examples. Source links identify the released datasets or environments, and teacher names follow the provenance records. We mark unknown upstream teachers as unspecified.

C.1 Reasoning Tasks

Both Qwen2.5 students use 36,000 training trajectories (9,000 per domain) and 800 validation examples (200 per domain). All reasoning trajectories come from public releases; Table 4 lists their sources and teachers. The stored `token_length` is measured with the Qwen2.5-1.5B-Instruct tokenizer and training chat template and includes prompt, completion, and template tokens. We keep sequences of at most 8,192 tokens.

The released sources are GLM and Kimi cleaned trajectories (Jackrong, 2026a;b), Sky-T1 (NovaSky, 2025), OpenThoughts-4 agreed-answer math traces (Marin Community, 2026), Mixture-of-Thoughts (Hugging Face, 2025b), and community GPT-OSS traces (iAmBoosted, 2026). The GLM and Kimi releases derive from the original datasets by Kassadin88 and ianncity (Kassadin88, 2026; ianncity, 2026). Open R1 releases Mixture-of-Thoughts (Hugging Face, 2025a). Its math, code, and science sources are OpenR1-Math, CodeForces CoTs, and Llama-Nemotron (Lozhkov et al., 2025; Penedo et al., 2025; Bercovich et al., 2025). The 600 GPT-OSS trajectories use 477 prompts from OpenThoughts-114k (Guha et al., 2025) and 123 from OpenCodeReasoning (Ahmad et al., 2025); GPT-OSS-20B generated the responses (OpenAI, 2025).

The manifest records 11,260 verified-correct rows, 9,000 upstream-verified rows, 13,140 cleaned rows with unverified answers, 2,000 answer-agreement rows, and 600 GPT-OSS rows without verified references (477 without a reference and 123 marked unverifiable). Verification strength therefore varies by source. Length filtering only checks that sequences fit the context window; answer-quality labels are kept separately in the manifest.

Example from the training corpus. Below is a verified DeepSeek-R1 Math example from Mixture-of-Thoughts; the middle of its reasoning is omitted for space.

> **User.** At the cross-country race, 8 runners wore white sports shirts. 4 runners wore blue sports shirts. How many runners started the cross-country race?

Table 4: Reasoning training sources, teachers, and full-sequence token statistics. Validation examples are excluded.

Domain	Task-data source	Teacher	Train	Link	Min	Mean	Max
Math	GLM cleaned	GLM-5.1	450	source	1,371	5894.3	8,187
Math	Kimi cleaned	Kimi-K2.5	690	source	2,361	6561.5	8,191
Math	Sky-T1	QwQ-32B / STILL-2	2,400	source	269	3927.4	8,165
Math	OpenThoughts-4 math	Qwen3-235B-A22B	2,000	source	1,535	6658.6	8,192
Math	Mixture-of-Thoughts	DeepSeek-R1	3,460	source	468	4109.7	8,189
Code	Sky-T1	QwQ-32B / STILL-2	3,600	source	798	4898.2	8,191
Code	GPT-OSS traces	GPT-OSS-20B	600	source	637	3264.4	7,978
Code	Mixture-of-Thoughts	DeepSeek-R1	4,800	source	551	5148.1	8,191
Science	GLM cleaned	GLM-5.1	3,000	source	2,014	3611.5	8,186
Science	Kimi cleaned	Kimi-K2.5	3,000	source	1,367	3421.0	7,633
Science	Mixture-of-Thoughts	DeepSeek-R1	3,000	source	389	2061.0	8,168
General	GLM cleaned	GLM-5.1	3,000	source	132	2935.9	8,142
General	Kimi cleaned	Kimi-K2.5	3,000	source	122	3399.5	8,180
General	Sky-T1	QwQ-32B / STILL-2	3,000	source	165	2981.3	8,164
Total			36,000	–	122	3991.0	8,192

Table 5: Agentic training sources, teachers, and full-sequence token statistics. Validation examples are excluded.

Domain group	Task-data source	Teacher	Train	Link	Min	Mean	Max
Coding	LiveCodeBench v1	DeepSeek V4 Flash	1,282	source	155	683.0	4,333
Customer	Nemotron interactive	Upstream; unspecified	1,200	source	1,837	3972.1	8,534
	τ^2 retail/airline	DeepSeek V4 Flash	2,997	source	4,899	9127.8	22,431
Search	Nemotron search	Upstream; unspecified	600	source	3,892	18791.6	32,701
Tool Interaction	TMax-SFT	Qwen3.6-27B	595	source	1,661	5575.2	30,575
	OpenThoughts-Agent	Upstream; unspecified	387	source	949	2730.1	17,593
	Nemotron tool calling	Upstream; unspecified	1,191	source	87	4178.3	31,182
Total			8,252	–	87	6498.2	32,701

Teacher reasoning, excerpt. *So 8 white shirts plus 4 blue shirts. … 8 plus 4 equals 12.* **Final answer:** $\boxed{12}$.

C.2 Agentic Tasks

After token-level deduplication, the agentic corpus contains 8,252 training and 824 validation trajectories. Table 5 reports the stored `sequence_length` measured with the Qwen3-8B tokenizer and chat template. Lengths include user and system text, tool schemas, tool observations, and assistant turns. The corpus contains 53,623,153 full-sequence tokens and 7,278,365 supervised assistant tokens, with a 32,768-token limit. Teacher reasoning text is removed; actions, observations, final answers, and tool-call links are retained.

Locally collected τ^2 (Barres et al., 2025) and LiveCodeBench (Jain et al., 2024) trajectories use DeepSeek V4 Flash. We reuse published Nemotron (NVIDIA, 2025) and OpenThoughts-Agent (Raoof et al., 2026) trajectories; the local metadata leaves their individual teachers unspecified. TMax contributes published successful trajectories (Ivison et al., 2026) with reasoning text removed. Final packaging removes five exact token-sequence duplicates. The 1,282 coding trajectories cover 296 unique prompts.

Example from the training corpus. The following LiveCodeBench training example uses a DeepSeek V4 Flash response. We excerpt the prompt and show the complete assistant code.

> **User, excerpt.** You are given an integer N between 1 and 9, inclusive, as input. Concatenate N copies of the digit N and print the resulting string.
>
> **Assistant.**

```
N = int(input())
print(str(N) * N)
```

D Evaluation Tasks and Splits

D.1 Reasoning Tasks

The ID suite contains four task families. OpenR1-Math contributes 300 mathematical problems. APPS contributes 250 programming problems in the benchmark's difficulty proportions: 50 introductory, 150 interview, and 50 competition tasks. Science MCQ contains 1,000 multiple-choice science questions selected from Mixture-of-Thoughts. RiddleSense contributes 1,018 multiple-choice commonsense riddles.

The OOD suite tests the same four capabilities using different question sources. GSM8K contains 1,319 grade-school mathematics word problems. MBPP+ contains 378 short Python programming tasks. ARC-Challenge contributes 608 multiple-choice science questions. BoolQ contributes 708 deduplicated yes/no questions about short passages, drawn from its labeled validation split. Table 1 summarizes the ID–OOD pairing for each domain.

D.2 Agentic Tasks

The ID suite covers customer interaction, coding, search, and tool interaction. The 60 τ^2 tasks are held-out retail and airline customer-service scenarios involving orders, returns, flights, and bookings. LiveCodeBench contributes 694 programming problems. Search ID contains 500 held-out Nemotron questions that call for finding and combining information. TMax contributes 300 tasks in terminal environments, where the agent works with commands and files.

The OOD suite uses different task sources or domains. Its 300 τ^2 cases come from telecom customer support and banking-knowledge retrieval. BigCodeBench contributes 2,280 evaluation cases: 1,140 software-oriented coding tasks presented in both completion and instruction formats. BFCL V4 contributes 5,106 function-calling cases, including tool selection, argument construction, and multi-turn interactions. FRAMES contains 824 factual questions that require retrieving and combining evidence from multiple sources.

E Token-Level Allocation Diagnostics

Figure 7 reconstructs token weights from saved Base probabilities and selected domain budgets for all 36,000 trajectories at each reasoning scale and 8,252 agentic trajectories. The 8B sources are pooled into four domain groups by supervised-token counts. Weights decrease with Base confidence by construction. The empirical question is where the training data place tokens along this curve.

The adjustment is concentrated in a subset of tokens. Within 8B Tool Interaction, the Terminal source has $a_t > 0$ on 9.3% of tokens, while 87.0% already receive Base probability above 0.9. Tokens with $a_t = 0$ keep the Base distribution as their training target. The fraction of adjusted tokens also does not simply follow the budget: at 1.5B, Science uses $R^\star = 0.45$ and adjusts 54.7% of tokens, whereas Code uses $R^\star = 0.60$ and adjusts 48.2%. The budget controls the initial gold-logit gradient ratio, whereas the adjusted-token fraction depends on Base confidence.

The same budget can likewise produce different adjustment rates. The three non-coding agentic groups all use $R^\star = 0.3$, yet their rates range from 16.9% on Search to 19.8% on Customer; Tool Interaction lies between them at 18.5%. Its Terminal and Tool Calling sources differ more sharply, at 9.3% and 25.5%. Thus, the distribution of Base confidence shapes how each group and source

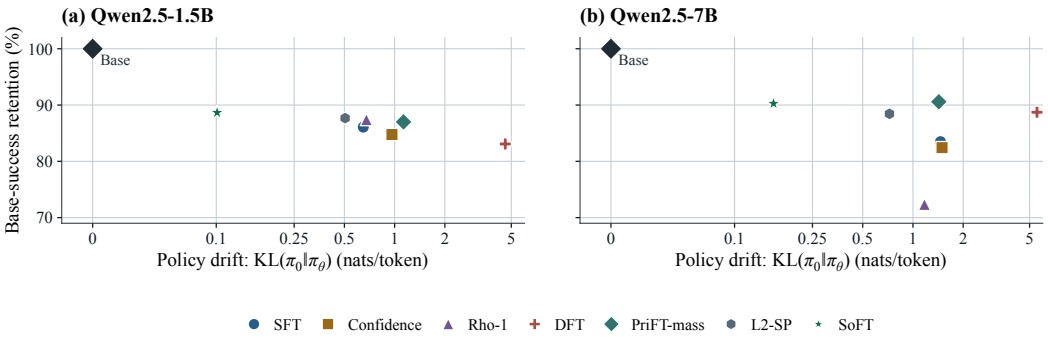

Figure 8: Final policy drift versus Base-success retention on reasoning tasks. Drift is $\mathrm{KL}(\pi_0\|\pi_\theta)$ over held-out teacher-forced prefixes; retention is the equal-weight mean over eight benchmarks.

receives supervision. Together, these measurements show how SoFT adapts token-level learning within a mixed training corpus.

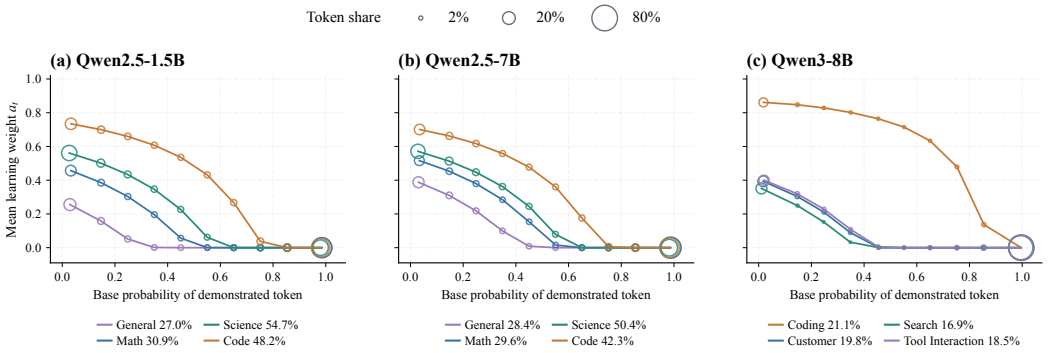

Figure 7: Token-weighted target allocation from Base caches. Lines: mean a_t by Base probability; circle areas: token fractions; legend percentages: fraction with $a_t > 0$. The 8B panel pools source data into four domain groups.

F Policy Drift and Behavioral Retention

Figure 3 normalizes each method's forward KL $\mathrm{KL}(\pi_0\|\pi_\theta)$ by that of SFT in the same setting. The reasoning measurements use the same 32 held-out validation trajectories per domain as RQ2 and average over the first 128 demonstrated tokens. The 8B measurement uses 16 validation trajectories from each of five agentic source domains and the first 128 supervised assistant tokens. Normalization places all settings on a common axis, although absolute KL still depends on the data.

Figure 8 compares absolute reasoning KL with mean Base-success retention. SoFT has the lowest drift and among the highest retention at both scales. At 1.5B, drift and retention broadly move in opposite directions. At 7B, DFT moves far from Base while retaining many successes, whereas Rho-1 moves less and retains fewer. Global KL captures one aspect of preservation; task success reveals another. The policy diagnostic uses validation prefixes, while retention is measured on the separate benchmark tasks.

G WHY NORMALIZED TARGET-KL CURVES CAN ALIGN

Section 6.3 shows nearly aligned normalized curves alongside different absolute target KL values. We use the KL chain rule, a mixture decomposition, and a low-probability limit to explain how this pattern can arise from the constructed targets.

G.1 REDUCING THE FULL-VOCABULARY KL TO A SCALAR

Write $r \in [0,1]$ for a candidate value of $R^\star$. For trajectory x, let T_x be the number of supervised positions. Assume $\pi_0(y_t^\star \mid s_t) > 0$ and $\sum_t (1 - \pi_0(y_t^\star \mid s_t)) > 0$. The floor $\tau_x(r)$ satisfies

$$\sum_t \max\{\tau_x(r) - \pi_0(y_t^\star \mid s_t), 0\} = r \sum_t (1 - \pi_0(y_t^\star \mid s_t)). \tag{25}$$

At the endpoints, use the Base target for $r = 0$ and the one-hot target for $r = 1$. Define the binary KL and binary entropy, using natural logarithms, as

$$\begin{aligned} b(u\|p) &= u \log \frac{u}{p} + (1-u) \log \frac{1-u}{1-p}, \\ h(u) &= -u \log u - (1-u) \log(1-u), \end{aligned} \tag{26}$$

with the usual continuous endpoint conventions and $b(1\|1) = 0$.

Proposition 1 (binary reduction). For the SoFT target, written $q_{\tau_x,t}$ in this section to make its dependence on the floor explicit,

$$\mathrm{KL}(q_{\tau_x,t} \,\|\, \pi_0(\cdot \mid s_t)) = b(\max\{\pi_0(y_t^\star \mid s_t), \tau_x\} \,\|\, \pi_0(y_t^\star \mid s_t)) \,. \tag{27}$$

Proof. Partition the vocabulary into the demonstrated token and its complement. The KL chain rule separates the divergence into the binary mass-allocation term and a conditional KL over other tokens. SoFT preserves the Base distribution conditional on that complement, so the conditional KL is zero. When $\pi_0(y_t^\star \mid s_t) = 1$, the target remains unchanged and both sides are zero.

The per-trajectory absolute and normalized changes are therefore

$$\begin{aligned} D_x(r) &= \frac{1}{T_x} \sum_t b(\max\{\pi_0(y_t^\star \mid s_t), \tau_x(r)\} \,\|\, \pi_0(y_t^\star \mid s_t)) \,, \\ C_x(r) &= \frac{D_x(r)}{D_x(1)}, \qquad D_x(1) = \frac{1}{T_x} \sum_t -\log \pi_0(y_t^\star \mid s_t). \end{aligned} \tag{28}$$

We form each normalized domain curve by averaging $C_x(r)$ over trajectories. The endpoints satisfy $C_x(0) = 0$ and $C_x(1) = 1$ by construction. The intermediate shape depends on the trajectories' Base probabilities and can vary across domains.

G.2 AN EXACT CANCELLATION OF TOKEN PROPORTIONS

Let P be the Base probability of the demonstrated token at a uniformly chosen position in a trajectory. Consider the idealized mixture

$$P \sim (1-f)\delta_1 + fF, \qquad 0 < f \le 1, \tag{29}$$

where δ_1 is a point mass at probability one and F is a distribution on $(0,1)$ with finite, positive $\mathbb{E}_F[-\log P]$. The first component represents tokens with full Base confidence; the second contains the remaining tokens. The point mass at one makes the cancellation explicit; in finite-logit models, high-confidence probabilities only approach one.

Proposition 2 (mixture-proportion invariance). For fixed F, the entire normalized curve is independent of f.

Proof. Both sides of the budget equation contain the same factor f:

$$f\, \mathbb{E}_F[(\tau - P)_+] = r f\, \mathbb{E}_F[1 - P]. \tag{30}$$

After canceling f, the floor depends only on F and r. Denote it by $\tau_F(r)$. The certain tokens contribute zero KL, so

$$\begin{aligned} D_x(r) &= f\,\mathbb{E}_F[b(\max\{P, \tau_F(r)\}\|P)], \\ D_x(1) &= f\,\mathbb{E}_F[-\log P], \\ C_x(r) &= \frac{\mathbb{E}_F[b(\max\{P, \tau_F(r)\}\|P)]}{\mathbb{E}_F[-\log P]}. \end{aligned} \tag{31}$$

The final expression contains no f.

For example, increasing the proportion of uncertain tokens from 10% to 50% multiplies absolute KL by five while leaving the normalized curve unchanged when F stays fixed. Domains can therefore require different amounts of adjustment yet have identical normalized responses. The result extends to domain averages when they share the same distribution of trajectory-level F values, even if their f values differ. The full vocabulary distributions may still differ.

G.3 A CLOSED FORM AND A LOW-PROBABILITY LIMIT

For a simple special case, set $F = \delta_\epsilon$ with $0 < \epsilon < 1$. All demonstrated tokens whose Base probability is below one then have probability ϵ. Equation 25 gives

$$\tau_\epsilon(r) = \epsilon + (1-\epsilon)r, \qquad C_\epsilon(r) = \frac{b(\epsilon + (1-\epsilon)r\|\epsilon)}{\log(1/\epsilon)}. \tag{32}$$

The proportion f cancels, while the confidence level ϵ continues to shape the curve.

Proposition 3 (entropy bound). For this special case and every $r \in [0, 1]$,

$$\max\left\{0, r - \frac{\log 2}{\log(1/\epsilon)}\right\} \le C_\epsilon(r) \le r. \tag{33}$$

Consequently, $C_\epsilon(r)$ converges uniformly to r as $\epsilon \to 0$.

Proof. Put $u = \epsilon + (1-\epsilon)r$ and $L = \log(1/\epsilon)$. Expanding the binary KL gives the exact identity

$$C_\epsilon(r) = u - \frac{h(u)}{L} + \frac{(1-u)\log(1/(1-\epsilon))}{L}. \tag{34}$$

Since $u \ge r$, $h(u) \le \log 2$, and the last term is nonnegative, this yields the lower bound, together with nonnegativity of KL. For the upper bound, $\mathrm{Bern}(u) = (1-r)\,\mathrm{Bern}(\epsilon) + r\delta_1$. Convexity of KL in its first argument gives $b(u\|\epsilon) \le r\log(1/\epsilon)$. The resulting uniform error bound is $0 \le r - C_\epsilon(r) \le \log 2/L$.

For sufficiently small ϵ, the leading term approaches r. The entropy term places the curve below the diagonal. The approximation remains sensitive to confidence: at $r = 0.5$, $C_\epsilon(r)$ is about 0.189 for $\epsilon = 0.5$ and 0.356 for $\epsilon = 0.01$. Thus, curve alignment also depends on the probability profile.

G.4 NEARLY CERTAIN TOKENS

For a more realistic mixture, replace δ_1 by a distribution G supported on $[1-\delta, 1]$. Let F be supported on $(0, \epsilon]$ and suppose the floor lies between ϵ and $1-\delta$. Then all F tokens are active and all G tokens are inactive. Define

$$\begin{gathered} \kappa = \frac{1-f}{f}, \qquad \mu = \mathbb{E}_F[P], \qquad \nu = \mathbb{E}_G[1-P], \\ \ell_F = \mathbb{E}_F[-\log P], \quad c_F = \mathbb{E}_F[-\log(1-P)], \quad \ell_G = \mathbb{E}_G[-\log P]. \end{gathered} \tag{35}$$

The budget equation and normalized divergence now give

$$\begin{aligned} \tau(r) &= r + (1-r)\mu + r\kappa\nu, \\ C_x(r) &= \frac{\tau(r)\ell_F - h(\tau(r)) + (1-\tau(r))c_F}{\ell_F + \kappa\ell_G}. \end{aligned} \tag{36}$$

Substituting the mixture into the budget equation and expanding binary KL over the active component gives these identities. The high-confidence component contributes residual gradient mass through $\kappa\nu$ and endpoint KL through $\kappa\ell_G$, restoring dependence on f. For fixed F and f, the idealized result emerges as G concentrates at one under the stated active-set condition. When κ is large, even small residuals from many nearly certain tokens can affect the curve.

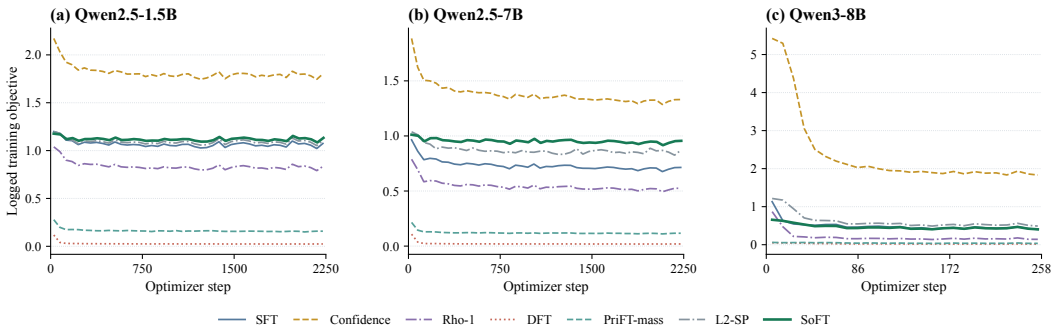

Figure 9: Training objectives averaged over 50-step (reasoning) or 10-step (agentic) windows. Objective scales vary across methods; benchmark scores report task performance.

G.5 A COUNTEREXAMPLE AND IMPLICATIONS FOR THE OBSERVATION

The homogeneous case has a specific bound. To see how a mixed probability profile differs, consider equal numbers of tokens with probabilities ϵ and $1/2$, and set $r = 1/2$. Both groups are active at

$$\tau = \frac{5}{8} + \frac{\epsilon}{4}. \tag{37}$$

As $\epsilon \to 0$, the low-probability group dominates both KL expressions, giving

$$C_x(1/2) = \frac{b(\tau\|\epsilon) + b(\tau\|1/2)}{\log(1/\epsilon) + \log 2} \longrightarrow \frac{5}{8}. \tag{38}$$

In contrast, the homogeneous low-probability case in Proposition 3 approaches $1/2$. Here, the normalized KL exceeds the budget, showing how different probability profiles can separate the curves.

The alignment in Section 6.3 is consistent with domains sharing a similar conditional probability profile while differing in the amount of required adjustment. The derivation identifies those profiles and their KL contributions as the quantities that shape the normalized curves.

H TRAINING DYNAMICS

The following curves cover the 21 trained runs in the main comparisons. Reasoning and agentic runs reach 2,250 and 258 steps, respectively; Base contributes only to evaluation. We average logged training metrics in non-overlapping 50-step and 10-step windows. Held-out entropy and learning rates use their recorded points.

H.1 TRAINING OBJECTIVES AND PREDICTIVE ENTROPY

Figure 9 shows distinct objective scales. Weighted losses are small for DFT and PriFT-mass, while Confidence reports larger values. SoFT's reasoning loss changes little because it includes the entropy of the soft target as a constant offset. The curves describe optimization within each run; benchmark results provide the performance comparison.

Figure 10 uses held-out entropy for reasoning and training entropy for agentic tasks. Final reasoning entropy is higher for SoFT than SFT: 1.228 versus 1.066 at 1.5B and 1.009 versus 0.713 at 7B. Over the last 10% of agentic steps, their means are similar (0.395 versus 0.386), while Confidence reaches 0.722 and DFT 0.066. SoFT's entropy therefore depends on the setting. Entropy measures how concentrated predictions are; correctness and Base preservation require separate evaluations. The different populations also limit comparisons across panels.

H.2 GRADIENT NORMS AND LEARNING-RATE SCHEDULES

SoFT's mean logged gradient norm decreases between the first and last 10% of steps: 0.770 to 0.555 at 1.5B, 1.659 to 0.644 at 7B, and 1.014 to 0.178 at 8B (Figure 11). These values describe the logged

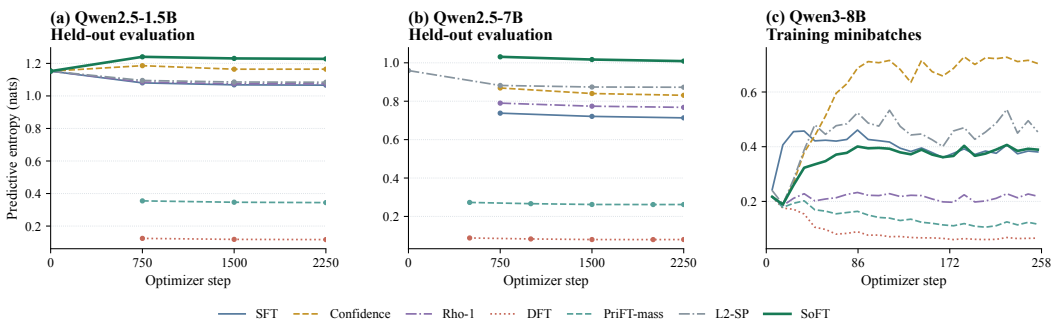

Figure 10: Predictive entropy (nats). Left/middle: recorded held-out evaluations. Right: training entropy in 10-step windows. Populations differ across panels; missing points are not imputed.

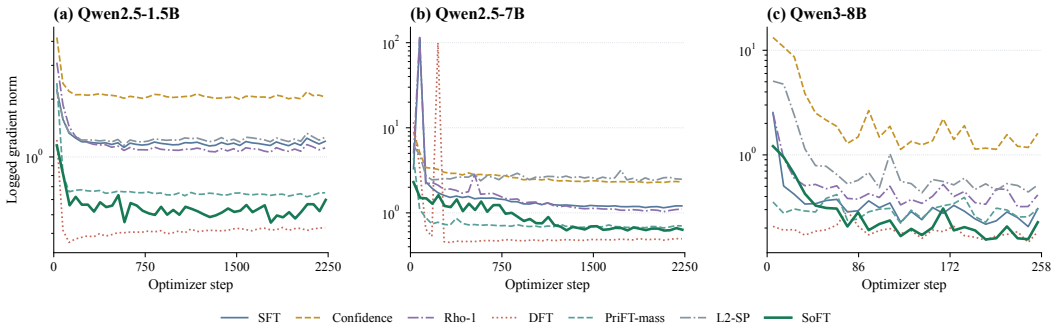

Figure 11: Logged gradient norms (log scale), using the windows in Figure 9. Magnitudes depend on objective scaling and trainer conventions.

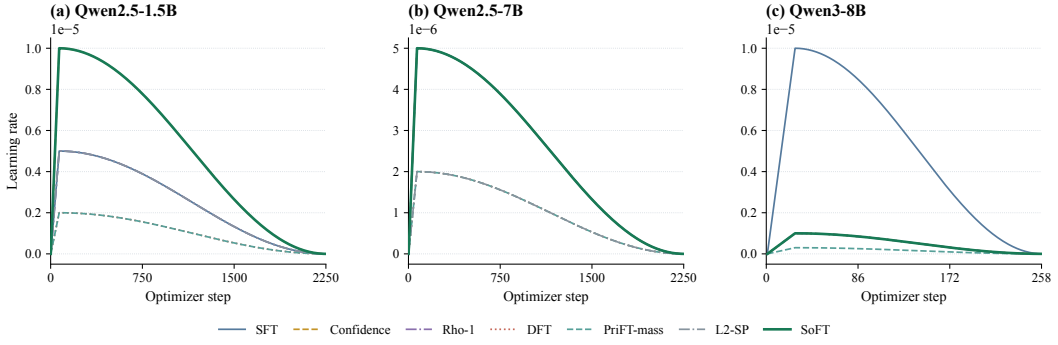

Figure 12: Recorded learning-rate schedules without smoothing. Identical schedules overlap.

update scale, and their interpretation depends on loss scaling, trainer conventions, and fluctuations hidden by window averages.

Figure 12 shows SoFT's peak rates of 10^{-5}, 5×10^{-6}, and 10^{-6}, followed by decay. Other methods use their own selected rates, and identical schedules overlap. Together, these curves characterize the optimization schedules chosen for the main comparisons.

I REALIZED DEMONSTRATION LEARNING

For each Qwen2.5 scale, we select 32 validation examples per reasoning domain by a fixed hash, using prompts of at most 512 tokens and at least 128 demonstrated tokens. All seven trained methods are scored on the same teacher-forced states and the first 128 demonstrated tokens. We report the gold-token probability gain relative to Base's remaining probability mass, as defined in Section 6.2. Horizontal bars in Figure 5 show approximate 95% ranges from 2,000 domain-stratified resamples of the selected trajectories; vertical coordinates are the main-table benchmark GMs. The prefix measurements use validation data, while the task scores come from the separate evaluation suites.

At 7B, three methods with almost the same aggregate acquisition ratio occupy widely separated positions on the vertical axis. Similar gold-token gains can therefore accompany different task outcomes. The scalar summarizes teacher-forced probability gain, while the retention analysis and benchmark scores capture behavior on complete responses.

Table 6: Realized acquisition by reasoning domain (%). Values use the same 32 validation examples per domain as Figure 5.

	Qwen2.5-1.5B			Qwen2.5-7B		
Domain	SFT	DFT	SoFT	SFT	DFT	SoFT
General	20.6	41.6	4.0	43.4	50.1	8.8
Math	24.9	47.2	6.0	47.2	52.9	11.3
Science	18.8	37.1	7.5	36.9	44.4	11.8
Code	18.9	40.0	10.2	38.4	44.2	16.0

The ordering DFT $>$ SFT $>$ SoFT holds in all four domains at both scales, so the aggregate contrast is not driven by a single domain. Within SoFT, Code has the largest realized gain at both scales and also receives the largest validation-selected budget. The domain budget and Base probability profile jointly shape the target used for these trajectories.