WHAT SHOULD SFT LEARN? CONDITIONAL INFORMATION GAIN FOR GENERALIZABLE POST-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Supervised fine-tuning (SFT) remains a widely used approach for post-training large language models, because fixed demonstrations can directly introduce desired reasoning strategies and behavioral patterns. Yet forcing a model to imitate externally provided trajectories may also interfere with behaviors already supported by pretraining, limiting generalization beyond the training distribution. Recent dynamic SFT methods mitigate this issue by reweighting token-level supervision according to model-side signals such as predictive probability, prior support, or distributional compatibility. However, *model compatibility is not equivalent to learning value*: a token that is weakly supported by the current model may either be an undesirable mismatch or precisely the task-specific decision that the demonstration is intended to teach. We propose **CIG-SFT**, an information-theoretic framework that estimates the task-conditioned information carried by each demonstrated token. Under teacher forcing, we compare the token likelihood from a frozen reference model with and without access to the task input while keeping the gold solution prefix fixed. The resulting conditional information gain quantifies how much the task itself reduces uncertainty about the demonstrated decision. We then convert this information into a KL-regularized projection of the current policy and derive an equivalent scalar-weighted SFT objective whose logit gradient exactly matches that of the projected target. Across mathematical reasoning and code-generation benchmarks, CIG-SFT consistently outperforms standard SFT and recent adaptive fine-tuning baselines, with particularly strong improvements in generalization to challenging evaluation distributions, while requiring neither additional training data nor online rollouts.[1]

1 INTRODUCTION

Large language models (LLMs) acquire broad and increasingly general capabilities during large-scale pretraining, ranging from language understanding and generation to reasoning and problem solving (Brown et al., 2020; Chowdhery et al., 2023). Since base models are not inherently optimized to express capabilities in the forms required by diverse downstream tasks, post-training plays a critical role in transforming a general-purpose base model into one that exhibits desired task- and application-specific behaviors (Chung et al., 2024). By providing explicit behavioral supervision or optimization signals, post-training can improve instruction following (Wei et al., 2021; Ouyang et al., 2022), shape how models reason and make intermediate decisions (Lightman et al., 2024; Guo et al., 2025), and encourage adherence to task-specific response structures and output conventions (Chu et al., 2025). These adaptations are particularly important for reasoning-intensive applications, where success depends not only on whether a model possesses the underlying capability, but also on whether it can invoke and organize that capability into an effective solution process. Accordingly, supervised fine-tuning (SFT) and reinforcement learning (RL) have become two central paradigms for modern LLM post-training, providing complementary mechanisms for shaping model behavior beyond pretraining (Ouyang et al., 2022; Schulman et al., 2017; Guo et al., 2025).

However, such adaptation introduces a fundamental tension between *acquiring target behaviors* and *preserving the capabilities that LLMs already support*. It is especially evident in post-training meth-

[1]Our code is available at an anonymous repository `https://anonymous.4open.science/r/CIG-SFT-7232/`.

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.

ods that learn from fixed, externally provided data, including SFT, offline preference optimization (Rafailov et al., 2023), and offline reinforcement learning (Snell et al., 2022). Their key advantage is behavioral coverage: by directly optimizing on curated demonstrations or feedback, these methods can expose the model to reasoning steps and behavioral patterns that it may rarely produce on its own. Yet this same property can also create a mismatch between externally specified supervision and the model's existing predictive behavior, biasing LLM behavior toward the demonstrated solution patterns at the expense of catastrophic forgetting of capabilities inherited from pretraining (Kotha et al., 2024; Li et al., 2024a). On-policy approaches address the same tension from the opposite direction, which generate training trajectories from the evolving policy itself and thus make the supervision naturally better aligned with LLMs' predictive behavior distribution, such as reinforcement learning (Schulman et al., 2017; Ouyang et al., 2022; Guo et al., 2025), self-taught reasoning (Zelikman et al., 2022), on-policy distillation (Agarwal et al., 2024), and on-policy self-distillation (Zhao et al., 2026). Nevertheless, this compatibility comes at the cost of behavioral coverage, as useful strategies encoded in external demonstrations may be missed when the current policy rarely visits them (Agarwal et al., 2024; Zhao et al., 2026).

Recent work mitigates this trade-off by retaining fixed demonstrations while selectively modulating their supervision at the token level. DFT reweights each target according to its probability under the current model (Wu et al., 2025), where subsequent methods stabilize this reweighting through distributional anchoring (Zhu et al., 2025), estimate token support from a frozen pretrained model (Wang et al., 2026b), or explicitly characterize whether a target is compatible with the model-induced distribution (Zhang et al., 2026). Despite different formulations, they share a common principle: *the strength of supervision is largely determined by how well a demonstrated token is supported by the model's predictive distribution*. However, predictive compatibility is not equivalent to learning value. A weakly supported target may indicate an undesirable distributional mismatch, but it may also correspond to precisely the task-specific decision that the demonstration is intended to teach. Conversely, a highly supported target may reflect a useful model-consistent behavior, or simply a trivial continuation already determined by the preceding gold trajectory. Model support alone cannot distinguish what it *should acquire* from what it *should avoid*, or what has been sufficiently captured.

Therefore, we quantify the learning value of each demonstrated token from an information-theoretic perspective, viewing information gain as the reduction in uncertainty induced by an additional observation (Shannon, 1948; Lindley, 1956; Cover & Thomas, 2006). In supervised fine-tuning, the task input specifies the behavior to be learned, while the preceding response tokens also contribute to predicting the next token. Model-support signals computed from the full context may conflate task-induced information with continuation-induced predictability. We disentangle these two sources by measuring the additional information contributed by the task input beyond the preceding context. In detail, we propose **CIG-SFT**, which translates pointwise conditional information gain into principled supervised updates. First, we estimate the task-conditioned information carried by each demonstrated token via comparing its log-likelihood ratio under the frozen base model with and without the task input while keeping the preceding response fixed. The resulting ratio is a pointwise conditional information quantity, which we normalize by task-free surprisal to obtain a bounded information-gain ratio. Next, we treat this value as a token-level reward and derive a KL-regularized projection of the current policy, producing an information-dependent target distribution that balances information acquisition against policy deviation. We show that the gradient induced by this projected target can be reproduced exactly by a scalar-weighted SFT objective. In this way, CIG-SFT provides a principled path from task-conditioned information to controlled policy adaptation, while retaining fixed demonstrations and requiring neither additional training data nor online rollouts.

We evaluate CIG-SFT on mathematical reasoning and code generation across multiple model scales. CIG-SFT consistently improves over standard SFT and strong adaptive fine-tuning baselines, while comparing favorably with reinforcement-learning methods without requiring additional data or online rollouts. Its gains persist across domains, training-set sizes, and hyperparameter settings, with efficiency in some low-data regimes. Ablation and training-dynamics analyses further show that CIG-SFT benefits from the separation between frozen information attribution and current-policy-aware proximal adaptation, producing more selective supervision and smaller parameter updates. Overall, our results demonstrate that task-conditioned information provides a useful complement to predictive compatibility for generalizable supervised post-training. Our contributions are threefold: (**i**) we identify task-conditioned information as a complementary signal to full-context model support for selective SFT; (**ii**) we derive an information-proximal target and show that its optimization

admits an exactly gradient-equivalent token-weighted objective; and (**iii**) we demonstrate consistent gains across model scales and domains, with data efficiency and more selective parameter updates.

2 Related Work

Supervised fine-tuning and adaptive objectives. Supervised fine-tuning (SFT) remains a standard component of LLM post-training, using dense token-level supervision on fixed demonstrations to adapt pretrained models to desired behaviors (Ouyang et al., 2022; Chung et al., 2024). Despite its effectiveness, directly fitting demonstrations can lead to overfitting, memorization, or degradation of capabilities outside the fine-tuning distribution (Kotha et al., 2024; Li et al., 2024a; Chu et al., 2025). Recent work revisits the uniform weighting of token-level supervision in standard SFT. Dynamic Fine-Tuning (DFT) rescales each target-token loss using its probability under the current model (Wu et al., 2025). Anchored SFT (ASFT) introduces KL-based anchoring to control distributional drift (Zhu et al., 2025). PriFT derives token weights from a frozen pretrained model to decouple prior support from the evolving optimization trajectory (Wang et al., 2026b), while IDFT uses distributional statistics to identify supervision better aligned with the model-induced distribution (Zhang et al., 2026). Related approaches such as ProFit and InfoSFT further explore probability- or information-based token selection and weighting (Liu et al., 2026; Sabbaghi et al., 2026). Collectively, these methods show that non-uniform supervision can improve the generalization of fixed-demonstration SFT. Our method, CIG-SFT, is closest to this line of work, but differs in the signal used to define token-level learning value. Existing adaptive SFT methods primarily derive their weights from predictive confidence, prior support, or distributional typicality under the complete context. CIG-SFT instead uses conditional information gain to quantify how much the task input contributes to each target beyond what the preceding context already reveals.

Reinforcement learning and on-policy post-training. Reinforcement learning provides a major paradigm for LLM post-training, where training trajectories are sampled from the evolving policy rather than fixed in advance. PPO-based RLHF has been widely used to optimize language models from preference-derived rewards (Schulman et al., 2017; Ouyang et al., 2022), while GRPO and subsequent variants enable scalable outcome-based optimization for reasoning tasks (Shao et al., 2024; Guo et al., 2025; Yu et al., 2026). Because supervision applies to states the current policy actually visits, on-policy optimization naturally matches the model's generative distribution and avoids forcing the model to imitate externally specified trajectories. However, the same policy dependence also limits behavioral coverage. Because on-policy supervision comes from trajectories generated by the current policy, behaviors the model rarely discovers are unlikely to receive strong learning signals. This matters especially when external demonstrations encode reasoning strategies or behavioral patterns not yet well represented in the current policy. Empirical studies of reasoning RL provide concrete manifestations of this limitation, including the absence of informative successful rollouts on difficult prompts and the progressive loss of exploration diversity during training (Cui et al., 2025; Yu et al., 2026; Wang et al., 2026a). On-policy methods improve the model's compatibility, but do not guarantee exposure to the behaviors that post-training aims to introduce.

3 Method

In this section, we present how to derive and implement CIG-SFT, an information-theoretic framework for selectively learning from fixed demonstrations. Our central principle is to separate two questions that are conflated in conventional token-weighted SFT: *(i) how much task-specific information does a demonstrated behavior contain, and (ii) given this information, how far should the current policy move toward that behavior?* CIG-SFT answers the first question through pointwise conditional information gain under a frozen reference model, and the second through a KL-regularized projection of the current policy. We further show that the resulting projected update can be realized exactly, at the logit-gradient level, by a scalar-weighted SFT objective.

3.1 Preliminaries

Let $\mathcal{D} = \{(x, y)\}$ denote a dataset of fixed demonstrations, where x is the task input and $y = (y_1, \ldots, y_T)$ is the response with T tokens. The preceding context for the t-th target token is denoted

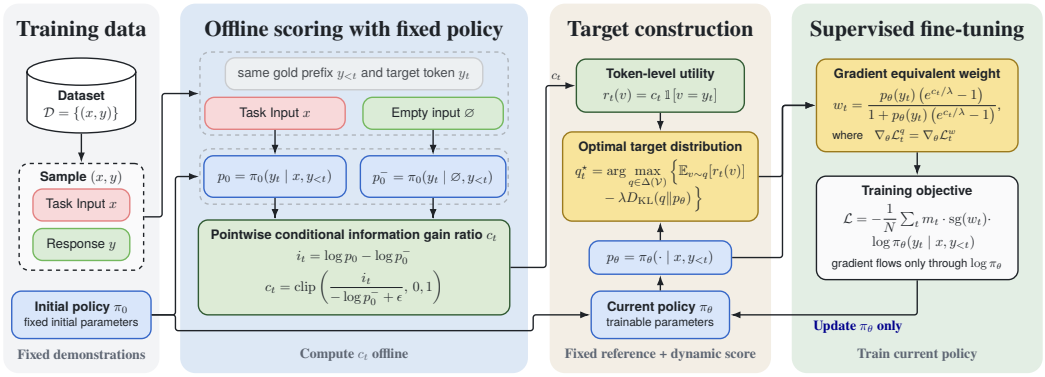

Figure 1: **Overview of CIG-SFT.** A frozen initial policy estimates token-level conditional information gain from fixed demonstrations. The information score and current policy jointly define an information-proximal target distribution, which is converted into a gradient-equivalent token weight for supervised optimization. During training, we update only the current policy.

as $y_{<t}$. We use π_0 to denote the frozen reference policy and π_θ the trainable policy initialized from the same pretrained model. The objective of standard supervised fine-tuning is to minimize:

$$\mathcal{L}_{\text{SFT}} = -\frac{1}{N}\sum_t m_t \log \pi_\theta(y_t \mid x, y_{<t}), \tag{1}$$

where $m_t \in \{0, 1\}$ masks tokens participating in training and $N = \sum_t m_t$. Dynamic or selective SFT methods generalize eq. (1) by assigning different learning strengths to individual target tokens:

$$\mathcal{L}_{\text{dyn}} = -\frac{1}{N}\sum_t m_t \ \text{sg}[\alpha_t] \log \pi_\theta(y_t \mid x, y_{<t}) + \beta \mathcal{R}(\theta), \tag{2}$$

where $\text{sg}[\cdot]$ denotes stop-gradient, $\mathcal{R}(\theta)$ optionally regularizes the updated policy with strength β, and α_t controls token-level supervision that is instantiated differently across methods:

$$\alpha_t = \begin{cases} \pi_\theta(y_t \mid x, y_{<t}) \\ u(\pi_0(\cdot \mid x, y_{<t}), y_t) \\ \pi_\theta(y_t \mid x, y_{<t})^{\gamma_t} \\ \pi_\theta(y_t \mid x, y_{<t}) \end{cases} \qquad \mathcal{R}(\theta) = \begin{cases} 0 & \text{DFT} \\ 0 & \text{PriFT} \\ 0 & \text{IDFT} \\ D_{\text{KL}}\left(\pi_0 \| \pi_\theta\right) & \text{ASFT} \end{cases} \tag{3}$$

where $u(\cdot)$ is a prior-support function computed from the frozen LLM, while γ_t is a distribution-compatibility coefficient determined by the target-token probability and the predictive uncertainty of the current policy. Despite their different instantiations, these methods share a common principle: token-level supervision is primarily determined by full-context model support, which does not disentangle the contribution of the task input x from that of the preceding context $y_{<t}$.

3.2 Pointwise Conditional Information Gain

To disentangle the influence of task evidence, we compare the same demonstrated token under task-conditioned and task-free contexts. Using the frozen reference policy π_0, we define

$$p_0 = \pi_0(y_t \mid x, y_{<t}), \qquad p_0^- = \pi_0(y_t \mid \varnothing, y_{<t}). \tag{4}$$

where $\varnothing$ denotes an empty task input under the same input format. The only difference between p_0 and p_0^- is the availability of task information x. This contrast is motivated by conditional mutual information (Cover & Thomas, 2006). For random variables X, Y_t, and $Y_{<t}$,

$$I(X; Y_t \mid Y_{<t}) = \mathbb{E}\left[\log \frac{p(Y_t \mid X, Y_{<t})}{p(Y_t \mid Y_{<t})}\right]. \tag{5}$$

Accordingly, inspired by the quantity inside the expectation of eq. (5), that is, the pointwise *conditional information gain*, we define our information gain i_t for each demonstrated token as

$$i_t = \log p_0 - \log p_0^- = \log \frac{\pi_0(y_t \mid x, y_{<t})}{\pi_0(y_t \mid \varnothing, y_{<t})}. \tag{6}$$

Since the exact marginal $\pi_0(y_t|y_{<t})$ is intractable, we use the task-free prediction $\pi_0(y_t \mid \varnothing, y_{<t})$ as an approximation (Xu et al., 2020; Ethayarajh et al., 2022). In information theory, the *surprisal* of an event with probability p is $-\log p$, which quantifies how unexpected the event is (Shannon, 1948). Therefore, i_t measures the reduction in target-token surprisal from observing x beyond what the preceding context already explains. We normalize this reduction by the task-free surprisal as

$$c_t = \mathrm{clip}\left(\frac{i_t}{-\log p_0^- + \epsilon}, 0, 1\right), \tag{7}$$

where $\epsilon > 0$ is a small constant for numerical stability when the task-free surprisal approaches zero. c_t quantifies the ratio of the target token's task-free surprisal explained by the task input: $c_t = 0$ indicates no positive surprisal reduction, while c_t approaching 1 indicates that the task input explains nearly all of the surprisal remaining after conditioning on the preceding context. Since c_t depends only on the frozen reference policy, it can be computed once and cached before fine-tuning.

3.3 Information-Proximal Projection

Conditional information gain quantifies task-side evidence for a demonstrated decision, but does not specify how far the current policy should move toward that decision. Motivated by reward-weighted policy optimization (Peters & Schaal, 2007; Wu et al., 2026; Zhu et al., 2025), we interpret the normalized information score c_t as a token-level utility for the demonstrated decision:

$$r_t(v) = c_t \, \mathbb{1}[v = y_t], \quad y_t \in \mathcal{V}, \quad \forall v \in \mathcal{V}, \tag{8}$$

where $\mathcal{V}$ denotes the LLM vocabulary. Instead of directly maximizing this utility, we constrain the update to remain close to the current policy, following regularized policy optimization (Peters et al., 2010; Schulman et al., 2015). Let $p_\theta = \pi_\theta(\cdot \mid x, y_{<t})$, we obtain the projected target distribution as

$$q_t^\star = \arg\max_{q \in \Delta(\mathcal{V})} \left\{ \mathbb{E}_{v \sim q}[r_t(v)] - \lambda D_{\mathrm{KL}}\left(q \| p_\theta\right) \right\}, \tag{9}$$

where $\Delta(\mathcal{V}) = \left\{ q : \mathcal{V} \to [0,1] \;\middle|\; \sum_{v \in \mathcal{V}} q(v) = 1 \right\}$ denotes the probability simplex over the vocabulary and $\lambda > 0$ controls the strength of the proximity constraint. The optimization in eq. (9) admits a closed-form solution through exponential tilting of the current policy (Peters et al., 2010),

$$q_t^\star(v) = \frac{p_\theta(v) \exp(r_t(v)/\lambda)}{Z_t}. \tag{10}$$

Since $r_t(v)$ is nonzero only for the demonstrated token y_t, the normalization constant simplifies to

$$\begin{aligned} Z_t &= \sum\nolimits_{v \in \mathcal{V}} p_\theta(v) \exp(r_t(v)/\lambda) = p_\theta(y_t) e^{c_t/\lambda} + \sum\nolimits_{v \in \mathcal{V} \setminus y_t} p_\theta(v) \\ &= p_\theta(y_t) e^{c_t/\lambda} + 1 - p_\theta(y_t) = 1 + p_\theta(y_t)\left(e^{c_t/\lambda} - 1\right), \end{aligned} \tag{11}$$

and the projected probabilities assigned to the target y_t and non-target token $v \neq y_t$ become

$$q_t^\star(y_t) = \frac{p_\theta(y_t) e^{c_t/\lambda}}{1 + p_\theta(y_t)\left(e^{c_t/\lambda} - 1\right)}, \quad \text{and} \quad q_t^\star(v) = \frac{p_\theta(v)}{1 + p_\theta(y_t)\left(e^{c_t/\lambda} - 1\right)}. \tag{12}$$

The projection adjusts the current policy according to the task-conditioned information score. When $c_t = 0$, $q_t^\star$ reduces exactly to the current policy, while increasing c_t or decreasing λ produces a larger shift toward the demonstrated decision that remaining proximal to the current policy.

3.4 Gradient-Equivalent Objective Realization

The projection yields the optimal target distribution $q_t^\star$ under eq. (9), which naturally serves as a soft supervision target for updating the trainable policy. We minimize the cross-entropy loss:

$$\mathcal{L}_t^q = -\sum\nolimits_{v \in \mathcal{V}} \mathrm{sg}[q_t^\star(v)] \log p_\theta(v). \tag{13}$$

Taking the gradient with respect to θ, we have

$$\nabla_\theta \mathcal{L}_t^q = -\sum\nolimits_{v \in \mathcal{V}} q_t^\star(v) \nabla_\theta \log p_\theta(v). \tag{14}$$

Since $p_\theta(v)$ is a normalized discrete distribution, there is

$$\sum\nolimits_{v\in\mathcal{V}} p_\theta(v)\nabla_\theta \log p_\theta(v) = \sum\nolimits_{v\in\mathcal{V}} \nabla_\theta p_\theta(v) = 0, \tag{15}$$

and hence

$$\sum\nolimits_{v\in\mathcal{V}\setminus y_t} p_\theta(v)\nabla_\theta \log p_\theta(v) = -p_\theta(y_t)\nabla_\theta \log p_\theta(y_t). \tag{16}$$

Using equations (12) and (16), we obtain

$$\nabla_\theta \mathcal{L}_t^q = -\left[\frac{p_\theta(y_t)(e^{c_t/\lambda}-1)}{1+p_\theta(y_t)(e^{c_t/\lambda}-1)}\right]\nabla_\theta \log p_\theta(y_t). \tag{17}$$

This gradient is exactly reproduced by the weighted hard-target objective

$$\mathcal{L}_t^w = -\,\mathrm{sg}(w_t)\log p_\theta(y_t),\ \text{as}\ \ \nabla_\theta \mathcal{L}_t^q = \nabla_\theta \mathcal{L}_t^w, \tag{18}$$

where the token-wise update coefficient w_t is

$$w_t = \frac{p_\theta(y_t)\left(e^{c_t/\lambda}-1\right)}{1+p_\theta(y_t)\left(e^{c_t/\lambda}-1\right)}. \tag{19}$$

It is worth noting that equations (13) and (18) are not generally equal in value. They are *gradient-equivalent* with respect to θ when their targets or weights are detached. Aggregating over all supervised tokens, we get the final training objective with equivalent gradients as

$$\mathcal{L}_{\text{CIG-SFT}} = -\frac{1}{N}\sum\nolimits_t m_t\ \mathrm{sg}(w_t)\log \pi_\theta(y_t \mid x, y_{<t}), \quad \text{where}\ N = \sum\nolimits_t m_t. \tag{20}$$

4 Experiments

4.1 Experimental Setup

Models and datasets. We evaluate CIG-SFT across mathematical reasoning and code generation. For mathematical reasoning, we use Qwen2.5-Math-1.5B and Qwen2.5-Math-7B (Yang et al., 2024), trained on a deduplicated 30K subset of NuminaMath-CoT (Li et al., 2024b). We evaluate on MATH500 (Lightman et al., 2024), Minerva Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), AIME 2024, AIME 2025, and AMC 2023[2]. For code generation, we use Qwen2.5-Coder-3B (Hui et al., 2024), trained on a 10K subset of the code training pool, and evaluate on HumanEval+ and MBPP+ (Liu et al., 2023), LiveCodeBench v5/v6 (Jain et al., 2025), BigCodeBench-Full (Zhuo et al., 2025), and DS-1000 (Lai et al., 2023).

Baselines. We compare with representative supervised fine-tuning, offline preference optimization, and on-policy reinforcement learning methods. For supervised fine-tuning, we include standard SFT and several recent adaptive objectives: Dynamic fine-tuning (DFT) (Wu et al., 2025), which reweights target tokens according to their probabilities under the current policy; In-distribution fine-tuning (IDFT) (Zhang et al., 2026), which combines target-token probability with predictive uncertainty to adapt the token-level learning signal; Anchored Supervised Fine-Tuning (ASFT) (Zhu et al., 2025), which augments probability-based reweighting with distributional anchoring; and PriFT (Wang et al., 2026b), for which we evaluate both PriFT-prob and PriFT-mass using token statistics from a frozen pretrained reference model. We additionally compare against offline preference optimization methods, including Direct Preference Optimization (DPO) (Rafailov et al., 2023) and SimPO (Meng et al., 2024), as well as on-policy reinforcement learning methods, including Group Relative Policy Optimization (GRPO) (Shao et al., 2024) and DAPO (Yu et al., 2026).

Training and evaluation settings. For mathematical reasoning, all supervised fine-tuning methods are trained for one epoch using AdamW (Loshchilov & Hutter, 2017), cosine learning-rate decay, a warm-up ratio of 0.1, a learning rate of 5×10^{-5}, a global batch size of 128, and a maximum sequence length of 2048. For CIG-SFT, we set $\lambda = 0.3$, which controls the strength of the conditional-information-gain signal when constructing the projected token-level target distribution. We keep this value fixed across all experiments unless otherwise specified. For evaluation, we sample 16 responses per problem at temperature 1.0 with a maximum generation length of 4096. We report Avg@16, the average correctness percentage across the 16 sampled responses, and Pass@16, the fraction of problems for which at least one of the 16 responses is correct.

[2] https://maa.org/maa-invitational-competitions/

Table 1: Mathematical reasoning results with Qwen2.5-Math-7B. We report Avg@16 / Pass@16.

Method	Average	MATH500	Minerva	Olympiad	AIME24	AIME25	AMC23
Base	17.2/55.6	39.4/88.2	11.7/46.7	14.8/57.2	8.5/46.7	2.9/20.0	25.8/75.0
SFT	19.7/49.4	53.0/87.2	16.9/48.9	19.2/55.1	2.3/13.3	1.7/16.7	25.3/75.0
DFT	31.7/51.1	68.9/86.0	23.9/43.4	33.5/56.3	11.7/23.3	5.8/20.0	46.6/77.5
IDFT	30.1/49.9	69.3/85.0	25.7/42.6	34.1/53.5	6.7/23.3	3.1/20.0	41.7/87.5
ASFT	30.3/60.1	67.8/92.2	27.2/53.7	33.7/64.6	8.1/33.3	3.1/26.7	41.7/90.0
PriFT-prob	35.8/62.5	72.3/92.2	29.6/**54.8**	36.8/**65.3**	14.4/43.3	7.5/26.7	53.9/92.5
PriFT-mass	36.4/63.2	74.0/91.8	27.8/52.9	36.9/61.8	15.0/46.7	7.5/**33.3**	**57.3**/**92.5**
CIG-SFT	**38.5**/**63.5**	**75.2**/**92.4**	**30.3**/53.3	**39.4**/64.3	**19.8**/**53.5**	**10.6**/30.0	55.5/87.5

Table 2: Mathematical reasoning results with Qwen2.5-Math-1.5B. We report Avg@16 / Pass@16.

Method	Average	MATH500	Minerva	Olympiad	AIME24	AIME25	AMC23
Base	16.5/51.6	39.0/86.0	10.2/41.2	19.2/52.1	4.8/26.7	1.9/23.3	23.6/80.0
Off-policy Reinforcement Learning							
DPO	17.3/49.7	40.0/86.0	11.1/44.1	19.7/55.0	4.2/33.3	0.8/10.0	28.0/80.0
SimPO	17.0/50.9	41.2/86.0	10.9/43.8	19.6/52.4	4.2/26.7	1.7/16.7	24.2/80.0
On-policy Reinforcement Learning							
GRPO	28.0/56.2	63.2/90.4	22.7/51.1	29.1/**62.1**	7.5/30.0	4.8/23.3	40.9/80.0
DAPO	30.5/57.7	67.2/**91.6**	**24.9**/**52.5**	30.2/60.7	9.6/23.3	**5.8**/**33.3**	45.0/85.0
Supervised Fine-Tuning							
SFT	13.8/42.3	41.5/81.8	12.1/40.4	12.0/45.6	0.6/10.0	0.2/ 3.3	16.4/72.5
DFT	26.3/49.0	64.0/85.8	20.8/41.5	26.9/53.3	5.4/13.3	3.8/20.0	37.0/80.0
IDFT	25.2/48.9	63.1/82.8	20.9/41.9	27.0/53.5	5.4/23.3	4.2/26.7	30.8/65.0
ASFT	27.6/54.8	63.8/90.0	21.2/50.4	28.5/59.0	7.9/23.3	4.2/23.3	39.7/82.5
PriFT-prob	30.3/56.2	66.6/90.8	22.8/49.6	30.4/59.3	10.0/30.0	5.2/20.0	46.6/87.5
PriFT-mass	30.2/54.1	66.2/90.4	21.7/49.6	31.4/57.2	**10.0**/36.7	5.4/13.3	46.6/77.5
CIG-SFT	**31.5**/**58.6**	**68.4**/89.6	24.1/48.5	**32.1**/59.4	9.4/**40.0**	5.2/26.7	**50.0**/**87.5**

4.2 Main Results

Mathematical reasoning. Tables 1 and 2 summarize the mathematical reasoning results. CIG-SFT consistently achieves the strongest aggregate performance across both model scales. On Qwen2.5-Math-1.5B, CIG-SFT reaches 31.5% **Avg@16** and 58.6% **Pass@16**, improving over standard SFT by $17.7/16.3$ and over DFT by $5.2/9.6$, respectively. It also outperforms the strongest prior supervised baseline, PriFT-prob, by 1.2 points in Avg@16 and 2.4 points in Pass@16. The same trend becomes more pronounced on Qwen2.5-Math-7B, where CIG-SFT achieves **38.5% Avg@16** and **63.5% Pass@16**, exceeding PriFT-mass by $2.1/0.3$ points and DFT by $6.8/12.4$ points. The improvements are distributed across benchmarks of different difficulty, with particularly clear gains on MATH500, OlympiadBench, and AIME.

Comparison with reinforcement learning. On Qwen2.5-Math-1.5B, CIG-SFT also compares favorably with reinforcement-learning baselines under the same evaluation protocol. It surpasses GRPO by $3.5/2.4$ and DAPO by $1.0/0.9$, while substantially outperforming the offline preference-optimization baselines DPO and SimPO. Notably, CIG-SFT obtains these results using only fixed demonstrations, without online rollouts or policy-generated training trajectories.

Code generation. Table 3 shows that CIG-SFT's advantage extends beyond mathematical reasoning. On Qwen2.5-Coder-3B, CIG-SFT achieves the best overall performance among the supervised methods compared, reaching 35.4% **Avg@16** and 50.5% **Pass@16**. This improves over SFT by

Table 3: Code-generation results with Qwen2.5-Coder-3B. We report Avg@16 / Pass@16.

Method	Average	HumanEval+	MBPP+	LCB v5	LCB v6	BigCode	DS-1000
Base	30.9/49.4	40.9/**67.1**	49.5/74.1	18.0/28.3	17.1/26.9	35.9/57.2	23.7/42.9
SFT	32.8/49.0	44.8/62.2	53.2/74.3	18.6/28.2	17.7/26.9	38.2/59.0	24.2/43.2
DFT	34.2/49.3	43.8/62.2	54.6/73.3	21.1/29.7	20.0/27.8	39.9/58.5	26.2/44.7
ASFT	34.3/49.7	45.6/65.2	54.2/73.3	21.1/29.3	20.1/27.9	39.5/59.0	25.2/43.7
PriFT-prob	33.6/49.3	45.6/64.0	52.6/73.5	19.9/29.1	19.1/27.8	39.0/56.9	25.6/44.6
CIG-SFT	**35.4**/**50.5**	**46.5**/64.0	**55.0**/**74.1**	**22.0**/**30.5**	**21.1**/**29.0**	**41.0**/**59.3**	**26.9**/**46.4**

Table 4: Ablations on mathematical reasoning with Qwen2.5-Math-1.5B.

Method	Average	MATH500	Minerva	Olympiad	AIME24	AIME25	AMC23
SFT	13.8/42.3	41.5/81.8	12.1/40.4	12.0/45.6	0.6/10.0	0.2/ 3.3	16.4/72.5
CIG-SFT	31.5/58.6	68.4/89.6	24.1/48.5	32.1/59.4	9.4/40.0	5.2/26.7	50.0/87.5
w/o c_t	21.9/48.4	55.4/88.6	19.2/46.0	22.1/54.7	4.0/16.7	1.4/ 6.7	29.4/77.5
w/o D_{KL}	28.4/55.1	62.1/90.4	19.3/46.0	29.2/59.1	9.6/30.0	4.4/20.0	45.5/85.0
Online c_t	6.9/21.9	21.1/54.6	6.6/22.4	3.3/12.6	0.2/ 3.3	0.2/ 3.3	9.8/24.4
Frozen D_{KL}	30.5/55.4	64.7/89.4	22.7/47.8	31.2/60.1	11.9/30.0	5.6/20.0	46.7/85.0
Both variants	15.0/35.1	43.8/76.8	11.8/30.1	7.6/23.7	1.9/13.3	0.4/ 6.7	24.4/60.0

$2.6/1.5$, DFT by $1.2/1.2$, and the strongest baseline in aggregate performance by $1.1/0.8$. The gains span HumanEval+, MBPP+, LiveCodeBench, BigCodeBench, and DS-1000, suggesting that the benefit of conditional-information-guided supervision is not specific to mathematical reasoning.

Ablation study. Table 4 examines both component removal and alternative design choices on Qwen2.5-Math-1.5B. Removing the information ratio (w/o c_t, eq. 7) causes the largest degradation among the component ablations, reducing Avg@16/Pass@16 from $31.5/58.6$ to $21.9/48.4$, while removing the KL-proximal constraint (w/o D_{KL}, eq. 9) yields a smaller but consistent drop to $28.4/55.1$. These results indicate that task-conditioned information provides the primary supervision signal, while the proximal projection further improves how this information is incorporated into the current policy. The design variants further highlight the distinct roles of the frozen reference and current policies. Computing c_t online with the evolving policy (Online c_t) substantially degrades performance to $6.9/21.9$, suggesting that decoupling information estimation from model optimization is important for obtaining a stable attribution signal. By contrast, replacing the current-policy KL reference with the frozen reference (Frozen D_{KL}) causes only a moderate decline, although it remains inferior to the proposed current-policy projection. Applying both modifications (Both variants) together reaches only $15.0/35.1$. Overall, the results support our design of using a frozen policy for offline information attribution and the evolving policy for proximal target construction.

4.3 Analysis

Training scale. We study the effect of training-set size by varying the number from 1K to 100K. CIG-SFT consistently achieves the strongest aggregate performance across all data scales, while exhibiting substantially higher data efficiency than the competing objectives. Notably, with only 1K demonstrations, CIG-SFT already reaches 28.9% Avg@16 and 55.9% Pass@16, outperforming SFT, DFT, ASFT, and IDFT even when they are trained on 100K demonstrations, whose best results are $26.8/53.6\%$. Moreover, CIG-SFT remains remarkably stable as the training set expands: Avg@16 stays within $[28.9, 31.5]\%$ and Pass@16 within $[55.2, 58.5]\%$ from 1K to 100K demonstrations. In contrast, standard SFT and DFT rely more strongly on increasing the amount of training data, while several dynamic SFT baselines exhibit non-monotonic behavior as the dataset grows. These results suggest that CIG-SFT extracts useful supervision more efficiently from fixed demonstrations and retains its advantage across both low-data and large-scale fine-tuning regimes.

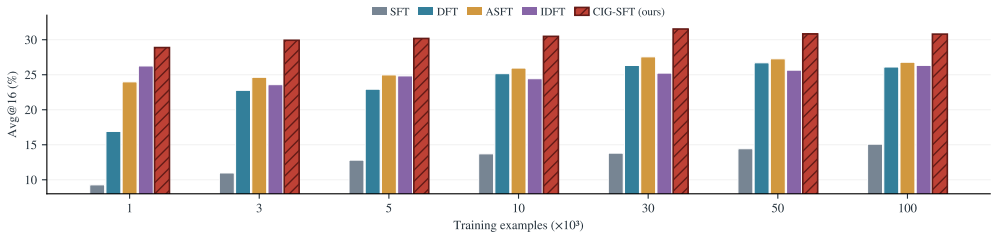

Figure 2: Analysis of training scale.

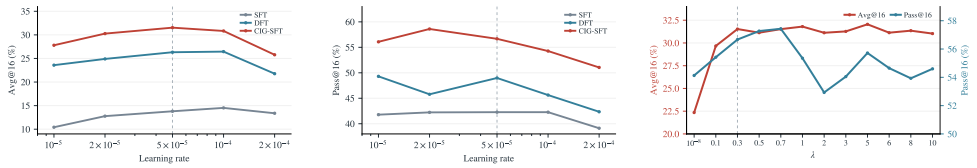

Figure 3: Analysis of learning rate and proximity strength λ.

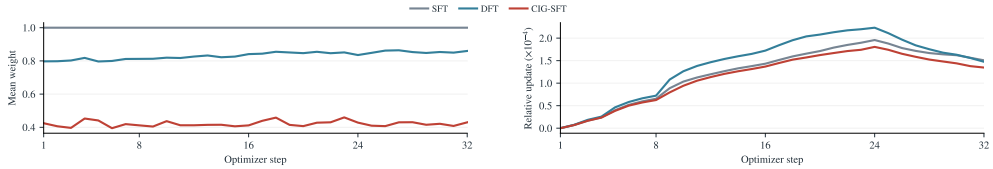

Figure 4: Analysis of the training dynamics.

Learning rate. We examine sensitivity to the learning rate over $\{1\times10^{-5}, 2\times10^{-5}, 5\times10^{-5}, 1\times10^{-4}, 2\times10^{-4}\}$. As shown in Figure 3, CIG-SFT consistently outperforms both SFT and DFT across the entire range on Avg@16 and Pass@16. Its performance remains relatively stable from 2×10^{-5} to 1×10^{-4}, with degradation occurring mainly at the largest learning rate. We use 5×10^{-5} in the main experiments, which provides a favorable balance between Avg@16 and Pass@16.

Proximity strength. We vary λ, which controls how strongly the projected target is constrained to remain close to the current policy. When λ is nearly zero, the proximity constraint effectively vanishes and performance drops substantially, highlighting the importance of controlling the information-induced policy shift. Once λ reaches a moderate value, Avg@16 remains stable over a broad range, with only minor fluctuations from 0.3 to 10. Pass@16 varies more but remains consistently strong in the same region. We use $\lambda = 0.3$ as the default setting in the main experiments.

Training dynamics. Figure 4 examines how different objectives allocate supervision and translate it into parameter updates. In the left panel, SFT assigns unit weight to every target token, while DFT maintains a relatively high average weight of approximately 0.8-0.86. In contrast, CIG-SFT consistently assigns substantially smaller average weights ($\sim$ 0.4-0.46), indicating more selective allocation of learning pressure across demonstrated tokens. To quantify the resulting optimization magnitude, we report the relative parameter update at optimizer step t, $\|\Delta\theta_t\|/\|\theta_t\|$, where $\Delta\theta_t$ denotes the actual parameter change produced by the t-th optimizer step. CIG-SFT induces consistently smaller relative parameter updates than DFT and generally smaller updates than standard SFT, while achieving stronger downstream performance. This suggests that its gains arise from more selective and better-targeted supervision rather than from more aggressive parameter movement.

5 Conclusion

We presented CIG-SFT, an information-theoretic framework that uses task-conditioned information to guide token-level supervision. It combines conditional information attribution with policy-aware proximal adaptation, yielding a simple gradient-equivalent objective. Across mathematical reasoning and code generation, CIG-SFT consistently outperforms recent dynamic fine-tuning methods while remaining competitive with reinforcement-learning baselines. The gains persist across learning rates and data scale, and strong performance can be obtained with substantially fewer demonstrations. Analysis shows that we assign lower average supervision weights and induce smaller relative parameter updates, suggesting that its benefit comes from more selective allocation of learning pressure rather than more aggressive fitting. Overall, these results highlight task-conditioned information as a useful complement to model-side compatibility for learning from fixed demonstrations.

AI USE STATEMENT

Generative AI tools are used as research and writing assistants during the development of this work. Specifically, they are used to assist with checking and reorganizing mathematical derivations, discussing experimental designs and interpretations, identifying potentially relevant literature, improving the clarity and structure of the manuscript, and assisting with the design of scientific figures. All AI-assisted technical content is independently reviewed and verified by the authors. The authors check mathematical derivations, manually verify cited literature against the original sources, and validate all reported experimental results and conclusions against the authors' own experiments. The authors take full responsibility for the final content of the paper, including all claims, equations, figures, and experimental results.

REPRODUCIBILITY STATEMENT

We provide the information necessary to reproduce CIG-SFT and its evaluation throughout the paper and supplementary materials. Section 3 specifies the complete training objective and its derivation, including the conditional information score, information-proximal projection, and gradient-equivalent objective. We report the models, datasets, baselines, evaluation metrics, and main hyperparameters used in our experiments. The training and evaluation code, together with the configuration files required to reproduce the reported experiments, is available in our anonymous repository: `https://anonymous.4open.science/r/CIG-SFT-7232/`.

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 *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

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

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of machine learning research*, 24(240):1–113, 2023.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of machine learning research*, 25(70):1–53, 2024.

Thomas M Cover and Joy A Thomas. Elements of information theory, 2/e. 2006.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. Understanding dataset difficulty with $\mathcal{V}$-usable information. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5988–6008. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/ethayarajh22a.html`.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Naman Jain, 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. In *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025.

Suhas Kotha, Jacob Springer, and Aditi Raghunathan. Understanding catastrophic forgetting in language models via implicit inference. In *International Conference on Learning Representations*, volume 2024, pp. 24110–24139, 2024.

Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Wen-tau Yih, Daniel Fried, Sida Wang, and Tao Yu. Ds-1000: A natural and reliable benchmark for data science code generation. In *International Conference on Machine Learning*, pp. 18319–18345. PMLR, 2023.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. Revisiting catastrophic forgetting in large language model tuning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4297–4308, Miami, Florida, USA, November 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.249. URL https://aclanthology.org/2024.findings-emnlp.249/.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9, 2024b.

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.

Dennis V Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, 1956.

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. *Advances in neural information processing systems*, 36:21558–21572, 2023.

Tao Liu, Taiqiang Wu, Runming Yang, Shaoning Sun, Junjie Wang, and Yujiu Yang. ProFit: Leveraging High-Value Signals in SFT via Probability-Guided Token Selection. *arXiv preprint arXiv:2601.09195*, 2026. URL https://arxiv.org/abs/2601.09195.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

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. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, pp. 745–750, 2007.

Jan Peters, Katharina Mulling, and Yasemin Altun. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pp. 1607–1612, 2010.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Mahdi Sabbaghi, George Pappas, Adel Javanmard, and Hamed Hassani. InfoSFT: Learn More and Forget Less with Information-Aware Token Weighting. *arXiv preprint arXiv:2605.14967*, 2026. URL https://arxiv.org/abs/2605.14967.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/schulman15.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Charlie Snell, Ilya Kostrikov, Yi Su, Mengjiao Yang, and Sergey Levine. Offline rl for natural language generation with implicit language q learning. *arXiv preprint arXiv:2206.11871*, 2022.

Chen Wang, Lai Wei, Yanzhi Zhang, Chenyang Shao, Zedong Dan, Weiran Huang, Ge Lan, and Yue Wang. Targeted exploration via unified entropy control for reinforcement learning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 16789–16802, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.828. URL https://aclanthology.org/2026.findings-acl.828/.

Ke Wang, Shuangqi Li, Mathieu Salzmann, and Pascal Frossard. PriFT: Prior-Support Guided Supervised Fine-Tuning. *arXiv preprint arXiv:2606.09396*, 2026b. URL https://arxiv.org/abs/2606.09396.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Yongliang Wu, Yizhou Zhou, Zhou Ziheng, 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. *arXiv preprint arXiv:2508.05629*, 2025. URL https://arxiv.org/abs/2508.05629.

Yongliang Wu, Yizhou Zhou, Ziheng Zhou, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, et al. On the generalization of sft: A reinforcement learning perspective with reward rectification. In *International Conference on Learning Representations*, volume 2026, pp. 27550–27571, 2026.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1eBeyHFDH.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Miaosen Zhang, Yishan Liu, Shuxia Lin, Xu Yang, Qi Dai, Chong Luo, Weihao Jiang, Peng Hou, Anxiang Zeng, Xin Geng, and Baining Guo. Towards On-Policy SFT: Distribution Discriminant Theory and its Applications in LLM Training. *arXiv preprint arXiv:2602.12222*, 2026. URL https://arxiv.org/abs/2602.12222.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

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, et al. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations*, volume 2025, pp. 66602–66656, 2025.



Figure 5: Token-level allocations for two representative cases. The columns show the demonstrated tokens, and the rows report SFT, DFT, task-conditioned information gain c_t, and update weights w_t.

A Appendix

A.1 Case Study: Task-Conditioned Credit and Model Probability

For each demonstrated answer token y_t, the frozen base model is evaluated under the original task input x and an erased-input counterfactual $\varnothing$, with the gold response prefix held fixed:

$$c_t = \mathrm{clip}\left(\frac{\log \pi_0(y_t \mid x, y_{<t}) - \log \pi_0(y_t \mid \varnothing, y_{<t})}{-\log \pi_0(y_t \mid \varnothing, y_{<t}) + \varepsilon}, 0, 1\right). \tag{21}$$

The resulting c_t values are detached and cached before training. We combine c_t with the current-policy probability to obtain w_t. Figure 5 qualitatively illustrates how CIG-SFT redistributes supervision compared with probability-based weighting. In Case A, task-relevant tokens such as *substitute*, *check*, the comparison operators, and the final option receive substantial task-conditioned information gain and are consequently assigned larger weights by CIG-SFT. In contrast, highly predictable continuation tokens such as *is* and *boxed* receive near-unit weights under DFT but zero task gain, and are therefore suppressed by CIG-SFT. Case B exhibits the same pattern: tokens tied to problem entities and arithmetic decisions, such as *sneakers*, *needed*, *action*, and *charged*, are emphasized, whereas generic continuation and formatting tokens are downweighted. These examples highlight the distinction between predictive support and task-conditioned information: **CIG-SFT preferentially allocates supervision to decisions explained by the task input rather than to tokens that are merely easy continuations of the gold prefix**.

A.2 Computational Cost

We characterize computational cost using both model-pass accounting and end-to-end measurements. We distinguish trainable forward passes, backward passes, and method-specific preprocessing, since their practical costs can differ substantially. For a fixed dataset $\mathcal{D}$, the training cost is

$$T_{\mathrm{train}}(\mathcal{D}) = T^{\theta}_{\mathrm{fwd}}(\mathcal{D}) + T^{\theta}_{\mathrm{bwd}}(\mathcal{D}) + T_{\mathrm{opt}}(\mathcal{D}) + T_{\mathrm{aux}}(\mathcal{D}). \tag{22}$$

Standard SFT requires one trainable forward-backward pass over each training batch. DFT and CIG-SFT reuse the target probabilities from this forward pass and therefore introduce only token-

Table 5: Measured training cost.

Method	s/Step	Peak GiB	Wall clock (h)
SFT	23.92	28.93	1.58
DFT	23.98	28.93	1.58
CIG-SFT	23.92	28.93	$0.48 + 1.58$
IDFT	25.61	42.84	1.68
ASFT	31.51	40.49	2.07
PriFT-prob	30.73	31.81	2.03
PriFT-mass	32.45	31.80	2.14

Table 6: Stability analysis on Qwen2.5-Math-1.5B.

Method	42	43	44	45	46	Mean $\pm$ Std.	95% CI
SFT	13.8	13.9	13.9	14.1	14.3	14.01 ± 0.20	$[13.76, 14.26]$
DFT	26.3	26.7	26.2	26.0	26.7	26.38 ± 0.34	$[25.95, 26.80]$
IDFT	25.2	26.4	26.0	25.6	27.0	26.05 ± 0.69	$[25.19, 26.90]$
PriFT-prob	30.3	29.2	29.3	29.3	29.3	29.46 ± 0.46	$[28.89, 30.03]$
CIG-SFT	**31.5**	**30.5**	**30.6**	**31.0**	**31.4**	$\mathbf{31.00} \pm 0.46$	$[\mathbf{30.42}, \mathbf{31.57}]$

wise scalar operations during training. CIG-SFT additionally performs one-time offline information scoring with the frozen reference policy. Specifically, each fixed demonstration is evaluated once with the original task input and once with an empty task input:

$$T_{\text{preprocess}} = T^0_{\text{fwd}}(\mathcal{D}) + T^0_{\text{fwd}}(\mathcal{D}^-), \tag{23}$$

where $\mathcal{D}^-$ denotes the task-free version of the fixed demonstrations. These passes require no gradient computation or backward pass. The resulting c_t are cached and reused throughout fine-tuning.

We additionally profile the end-to-end cost on Qwen2.5-Math-1.5B using the same 30K-example training set as the main experiments. All methods use a global batch size of 128, micro-batch size of 4, maximum sequence length of 2048, and 233 optimizer steps. Peak GPU memory is measured using `torch.cuda.max_memory_allocated()`. As shown in Table 5, the online training cost of CIG-SFT is essentially identical to standard SFT, with the same peak GPU memory of 28.93 GiB. The one-time offline scoring stage takes an additional 0.48 hours, giving an end-to-end cost of approximately 2.06 hours when preprocessing is included. The cached information scores increase offline storage from 41.5 MiB to 254.2 MiB, an additional 212.7 MiB, while introducing no increase in peak training GPU memory.

A.3 Stability Across Random Seeds

We evaluate optimization stability across five random seeds (42-46) on Qwen2.5-Math-1.5B. As shown in Table 6, CIG-SFT consistently achieves the strongest performance across all seeds, ranging from 30.5% to 31.5% Avg@16. Its mean performance is 31.00 ± 0.46, with a 95% confidence interval of $[30.42, 31.57]$, indicating limited sensitivity to random initialization and data-order variation. Notably, the confidence interval remains above that of the strongest competing baseline, PriFT-prob ($[28.89, 30.03]$), suggesting that the observed improvement of CIG-SFT is robust rather than driven by a favorable random seed.

A.4 Detailed Data-Scale Results

Table 7 reports the complete numerical results underlying the training-scale analysis in Figure 2. We vary the number of training demonstrations from 1K to 100K and report Avg@16/Pass@16 for each method at every scale. These detailed results complement the aggregate trends shown in the main text and enable direct comparison across individual training-set sizes.

Table 7: Results of Avg@16 / Pass@16 across training-set sizes.

Method	1K	3K	5K	10K	30K	50K	100K
Base	16.5/51.6	16.5/51.6	16.5/51.6	16.5/51.6	16.5/51.6	16.5/51.6	16.5/51.6
SFT	9.3/38.4	11.0/38.6	12.8/41.2	13.7/43.4	13.8/42.3	14.4/42.5	15.1/42.7
DFT	16.9/45.5	22.8/46.2	22.9/46.1	25.2/49.8	26.3/49.0	26.7/46.9	26.1/47.5
ASFT	24.0/53.7	24.6/53.1	25.0/53.4	26.0/51.0	27.6/54.8	27.3/53.8	26.8/53.6
IDFT	26.3/**57.0**	23.6/47.6	24.8/45.9	24.4/44.4	25.2/48.9	25.7/46.1	26.3/45.3
CIG-SFT	**28.9**/55.9	**29.9**/**56.4**	**30.2**/**58.5**	**30.5**/**55.2**	**31.5**/**56.7**	**30.8**/**56.9**	**30.8**/**57.3**

Table 8: Detailed sensitivity results for the learning rate.

Learning rate	SFT	DFT	CIG-SFT
1×10^{-5}	10.4/41.8	23.6/**49.3**	27.8/56.1
2×10^{-5}	12.8/42.2	24.9/45.8	30.3/**58.6**
5×10^{-5}	13.8/**42.3**	26.3/49.0	**31.5**/56.7
1×10^{-4}	**14.5**/**42.3**	**26.5**/45.6	30.8/54.3
2×10^{-4}	13.4/39.1	21.8/42.4	25.8/51.1

A.5 Hyperparameter Sensitivity

Tables 8 and 9 provide the complete numerical results underlying the sensitivity analyses shown in Figures 3 and 4. Table 8 reports Avg@16/Pass@16 for SFT, DFT, and CIG-SFT across five learning rates, while Table 9 reports the corresponding CIG-SFT results under different values of the projection coefficient λ. These tables complement the trends summarized in the main text and provide the exact values used in the sensitivity plots. Consistent with the main-text analysis, CIG-SFT remains the strongest method across all tested learning rates. For the projection coefficient, performance drops sharply when λ is nearly zero, while remaining relatively stable over a broad range of moderate values.

A.6 General Capability Retention Analysis

We examine whether improving mathematical reasoning comes at the expense of the model's general capabilities. Table 10 reports accuracy on four general-purpose benchmarks before and after fine-tuning. CIG-SFT largely preserves the base model's performance, with the average accuracy changing only from 56.1% to 55.9%. The differences on individual benchmarks are also small: -0.2 on MMLU-General, -0.5 on ARC-Challenge, -0.6 on HellaSwag, and $+0.3$ on PIQA. In comparison, DFT and IDFT reduce the average accuracy to 54.9% and 54.4%, respectively, while ASFT reaches 55.7%. These results provide additional evidence that the reasoning improvements of CIG-SFT are obtained with limited degradation of the general capabilities.

A.7 In-Domain Generalization

We evaluate whether the advantage of CIG-SFT extends to unseen examples drawn from the same distribution as the training data. Specifically, we randomly sample 1,000 examples from the NuminaMath-CoT data source that are excluded from the training set. As shown in Table 11, all fine-tuning methods substantially improve over the base model, confirming that they effectively adapt to the training-domain reasoning distribution. CIG-SFT achieves the best Avg@16 of 51.8%, outperforming DFT (50.8%), ASFT (50.5%), and standard SFT (42.4%). These results indicate that the gains of CIG-SFT are not restricted to out-of-distribution benchmarks: selectively allocating supervision also improves generalization to unseen examples from the training distribution.

Table 9: Detailed sensitivity results for the projection coefficient λ.

λ	10^{-8}	**0.1**	**0.3**	**0.5**	**0.7**	**1.0**	**2**	**3**	**5**	**6**	**8**	**10**
Avg@16	22.4	29.7	31.5	31.1	31.5	<u>31.8</u>	31.1	31.3	**32.1**	31.1	31.4	31.0
Pass@16	54.1	55.4	56.7	<u>57.3</u>	**57.4**	55.4	52.9	54.1	55.7	54.7	53.9	54.6

Table 10: General-task performance. Values are raw Acc (%).

Method	**Average**	**MMLU-General**	**ARC-Challenge**	**HellaSwag**	**PIQA**
Base	**56.1**	<u>58.2</u>	**48.5**	**45.2**	72.6
SFT	<u>55.9</u>	57.9	<u>48.0</u>	44.6	**73.0**
DFT	54.9	57.9	47.1	42.4	72.0
IDFT	54.4	57.7	46.2	41.4	72.2
ASFT	55.7	**58.3**	47.5	44.5	72.6
CIG-SFT	<u>55.9</u>	58.0	<u>48.0</u>	<u>44.6</u>	<u>72.9</u>

Table 11: In domain generalization performance for Qwen2.5-Math-7B.

Method	**Avg@16**
Base	32.2
SFT	42.4
DFT	<u>50.8</u>
IDFT	49.7
ASFT	50.5
PriFT-prob	49.2
PriFT-mass	47.5
CIG-SFT	**51.8**