Online Self-Weighted Fine-Tuning

Haiquan Wen[1,*] **Yiwei He**[1,*] **Bei Peng**[2] **Guangliang Cheng**[1,†]
[1]University of Liverpool [2]University of Sheffield
[*]Equal contribution. [†]**Correspondence:** guangliang.cheng@liverpool.ac.uk

Abstract

Standard supervised fine-tuning (SFT) assigns the same explicit loss weight to every expert demonstration, regardless of the model's changing competence over training queries. Reinforcement learning (RL) based methods adapt update strength using model-generated rollouts, but often require substantially more sampling and can be unstable on hard tasks. We propose **Online Self-Weighted Fine-Tuning (OSW-FT)**, a simple method that augments SFT with online, trajectory-level weighting. For each query, OSW-FT estimates the model's current success rate using a small number of inference-only rollouts and rescales the standard SFT loss accordingly. The optimization direction remains anchored to the expert trajectory, while the update magnitude adapts online. For binary-verifiable reasoning, we connect this weighting to SFT and RL at the gradient level, inspired by variance-reduction principles. The resulting estimator is unbiased for the exact OSW-FT surrogate update for any finite rollout count, and we analyze convergence with respect to the corresponding surrogate objective. Evaluated across Qwen3 series ranging from 0.6B to 4B on multiple challenging benchmarks (e.g., AIME), OSW-FT consistently improves over SFT on small-to-medium scale models. OSW-FT offers a favorable compute-performance trade-off as a practical approach for fine-tuning small-to-medium LLMs on binary-verifiable reasoning tasks with only **2 online rollouts**.

1 Introduction

Post-training a LLM raises a deceptively simple question: *how much should each training sample update the model at its current stage of training?* This question is especially consequential for verifiable reasoning tasks, where the model's competence on individual training queries can vary substantially and evolves throughout training. Some queries may already be reliably solved by the model and provide limited additional learning signal, while others remain near its capability frontier and warrant concentrated optimization effort. Standard Supervised Fine-Tuning (SFT) (Ouyang et al., 2022), however, assigns a uniform sample weight to every expert demonstration throughout training, treating already-mastered and unresolved queries identically. While this design is simple and stable, it allocates optimization effort coarsely across examples with very different learning value.

Reinforcement learning (RL) based post-training methods (Guo et al., 2025; Schulman et al., 2017) address this issue from a different direction. By computing advantages from model-generated rollouts, they adapt update strength to the model's current behavior and can focus learning on more informative examples. This adaptivity is appealing, but it comes with substantial online sampling cost and optimization instability, especially on hard reasoning tasks where failing to discover any correct trajectories entirely deprives the model of a viable learning direction. SFT avoids this failure mode by always providing a high-quality expert trajectory as the supervision target, but lacks a mechanism to modulate how strongly that trajectory should be learned from. This suggests a natural middle ground: preserve the stable, expert-anchored optimization direction of SFT, while introducing a lightweight online signal that adjusts update magnitude according to the model's current competence.

We propose **Online Self-Weighted Fine-Tuning (OSW-FT)**, a simple post-training method for verifiable reasoning that changes how much to learn from each expert trajectory without changing what to learn from. For each training query, OSW-FT performs a small number of inference-only rollouts to estimate the model's current pass rate $\hat{p}(q)$, and reweights the standard SFT loss by a factor of $(1 - \hat{p}(q))$. This weight is largest for queries the model consistently fails on, thereby concentrating optimization effort near the capability frontier,

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 11872–11884
October 24-29, 2026 ©2026 Association for Computational Linguistics

and naturally decreases toward zero as the model masters a query. Crucially, OSW-FT uses online rollouts only to estimate this scalar weight; it does not optimize on model-generated trajectories, does not perform token-level reward shaping, and does not modify the training distribution. This design targets settings with a limited online rollout budget while retaining expert-trajectory supervision.

For binary-verifiable reasoning, we construct an explanatory framework connecting this weighting to RL control variates at the gradient level. OSW-FT restores this quantity online through a Monte Carlo estimate from a small number of rollouts. Under our analysis, the resulting estimator is unbiased for the exact OSW-FT surrogate update for any finite rollout count K, and the convergence analysis applies to the corresponding surrogate objective under the stated assumptions. The method is naturally most useful when the training distribution remains challenging for the current model; when most training queries are already solved, the online weight correspondingly diminishes.

We evaluate OSW-FT on Qwen3 series (Yang et al., 2025) models ranging from 0.6B to 4B parameters across AIME (AIME, 2025), AMC (AMC, 2023), MATH-500 (Lightman et al., 2023), and GPQA-Diamond (Rein et al., 2024). Empirically, OSW-FT consistently improves over standard SFT on small-to-medium model scales. Compared with GRPO (Shao et al., 2024), the results vary across models and metrics: OSW-FT performs favorably in several limited-rollout settings, whereas GRPO remains stronger in some configurations. OSW-FT is therefore not intended as a general replacement for RL; it targets binary-verifiable reasoning when the available online rollout budget is limited. Ablation studies further validate the benefit of continuous online success-rate weighting over alternative hard-thresholding or random heuristics.

In summary, this work makes the following core contributions: **First**, we introduce OSW-FT, an expert-trajectory-anchored method that modulates SFT updates using an online success-rate weight for verifiable reasoning. **Second**, we analyze finite-rollout unbiasedness and convergence with respect to the proposed surrogate update. **Third**, evaluations show that OSW-FT consistently improves over standard SFT on small-to-medium scales and provides a favorable compute–performance trade-off under a limited online rollout budget.

2 Online Self-Weighted Fine-Tuning

Supervised Fine-Tuning (SFT) aligns language models by maximizing the log-likelihood of expert demonstrations. Given a dataset $\mathcal{D} = \{(q, o^*)\}$ of input queries q and ground-truth output trajectories o^*, SFT trains the model π_θ by applying a uniform weight of 1 to the gradient of every ground-truth trajectory:

$$\begin{aligned} -\nabla_\theta \mathcal{J}_{\text{SFT}} &= \nabla_\theta \mathcal{L}_{\text{SFT}} \\ &= -\mathbb{E}_{(q,o^*)\sim\mathcal{D}}\left[1 \cdot \nabla_\theta \log \pi_\theta(o^* \mid q)\right] \end{aligned} \tag{1}$$

From a classical estimation perspective, this uniform weighting is strictly principled as the standard maximum likelihood estimator over the data distribution. However, from an online optimization perspective, applying full gradient magnitude to queries the model has already mastered ($\pi_\theta(o^* \mid q) \to 1$) becomes statistically inefficient; it injects redundant gradient noise regarding the evolving policy and dampens the learning efficacy on unresolved frontier queries. To derive a principled variance-reduced gradient estimator that dynamically modulates this update intensity, we cast the problem within the framework of score function estimation.

2.1 Score Function Decomposition and Optimal Control Variate

For verifiable reasoning tasks, let $R(o) \in \{0, 1\}$ denote the binary correctness of a generated trajectory. The alignment objective is to maximize expected correctness:

$$\mathcal{J}(\theta) = \mathbb{E}_{q\sim\mathcal{D}}\mathbb{E}_{o\sim\pi_\theta(\cdot|q)}[R(o)] \tag{2}$$

Applying the log-derivative trick and subtracting a query-dependent control variate $c(q)$—which leaves the expected gradient unbiased since $\mathbb{E}_{o\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(o \mid q)] = 0$—yields the standard variance-reduced score function estimator:

$$\begin{aligned} \nabla_\theta \mathcal{J} = \mathbb{E}_{q\sim\mathcal{D}}\mathbb{E}_{o\sim\pi_\theta}\Big[\\ (R(o) - c(q))\nabla_\theta \log \pi_\theta(o \mid q)\Big] \end{aligned} \tag{3}$$

We seek optimal control variate $c^*(q)$ that minimizes the MSE of the scalar learning signal:

$$\min_{c(q)} \mathbb{E}_{o\sim\pi_\theta}\left[(R(o) - c(q))^2\right] \tag{4}$$

Setting the derivative to zero yields the exact analytical solution $c^*(q) = \mathbb{E}_{o\sim\pi_\theta}[R(o) \mid q]$, which for binary tasks simplifies to the model's current success probability $p_s(q)$. Crucially, this optimal control variate is identical to the advantage of the oracle trajectory:

$$\begin{aligned} A(o^*, q) &\triangleq R(o^*) - \mathbb{E}_{o\sim\pi_\theta}[R(o) \mid q] \\ &= 1 - p_s(q) \end{aligned} \tag{5}$$

The MSE-minimizing baseline and the unbiased advantage estimator for o^* are therefore identical for binary rewards, providing a unified statistical justification for the signal magnitude derived below. Substituting $c^*(q) = p_s(q)$ and applying the Law of Total Expectation to decompose by reward outcome, we observe a **dual role** of $p_s(q)$: it represents both probability of generating correct trajectory ($\Pr[R = 1]$) and determines its advantage magnitude:

$$\begin{aligned} \nabla_\theta \mathcal{J} = \mathbb{E}_{q\sim\mathcal{D}}\Big[& \underbrace{p_s(q)}_{\Pr[R=1]} \cdot \mathbb{E}_{o\sim\pi_\theta|R(o)=1}\Big[\\ & \underbrace{(1 - p_s(q))}_{A(o^*,q)} \nabla_\theta \log \pi_\theta(o \mid q)\Big] \\ & + (1 - p_s(q)) \cdot \mathbb{E}_{o\sim\pi_\theta|R(o)=0}\Big[\\ & (0 - p_s(q))\, \nabla_\theta \log \pi_\theta(o \mid q)\Big]\Big] \end{aligned} \tag{6}$$

2.2 Oracle Substitution in SFT

The equation highlights a fundamental tension: the positive learning signal is gated by the success probability $p_s(q)$. In the low-resource or early-training regime where $p_s(q) \to 0$, **pure reinforcement learning methods** suffer from **gradient starvation**, as the agent cannot explore and discover valid reasoning paths from scratch.

SFT as Deterministic Oracle Substitution. The essence of SFT is to bypass this exploration bottleneck by replacing the intractable generative probability with the absolute certainty of the dataset $\mathcal{D}$. By manually setting the probability of encountering a correct trajectory strictly to unity ($\Pr[R = 1] \to 1$), SFT not only guarantees a continuous learning signal but also **mathematically nullifies the negative gradient term**:

$$\begin{aligned} \nabla_\theta \mathcal{J} \approx \mathbb{E}_{q\sim\mathcal{D}}\Big[& \underbrace{1}_{\text{Dataset Certainty}} \cdot \\ & \mathbb{E}_{o\sim\pi_\theta|R(o)=1}\left[\underbrace{(1 - p_s(q))}_{A(o^*,q)} \nabla_\theta \log \pi_\theta(o \mid q)\right]\Big] \end{aligned} \tag{7}$$

The oracle trajectory o^* serves as a deterministic surrogate for the intractable conditional expectation $\mathbb{E}_{o\sim\pi_\theta|R(o)=1}[\cdot]$, translating to: $\nabla_\theta \log \pi_\theta(o \mid q) \approx \nabla_\theta \log \pi_\theta(o^* \mid q)$.

Justification for Asymmetric Substitution. A technical subtlety arises in this transition: the success probability $p_s(q)$ is substituted by 1 in the empirical frequency slot ($\Pr[R = 1]$) but retained within the inner advantage term $(1 - p_s(q))$. We justify this asymmetric treatment through the lens of *distributional decoupling*:

- **Directional Alignment (Data-centric):** SFT performs a strict distributional shift, replacing the model's stochastic generative distribution $\pi_\theta(\cdot \mid R(o) = 1)$ with the deterministic expert dataset $\mathcal{D}$. In this expert domain, the empirical frequency of encountering an oracle trajectory is unity ($P_\mathcal{D}(R = 1) = 1$), which nullifies the negative gradient term and anchors the optimization direction to o^*.
- **Magnitude Regulation (Model-centric):** While the optimization *direction* is dictated by the static data distribution, the update *magnitude* must respect the variance-reduced baseline governed by the evolving policy. Retaining $(1 - p_s(q))$ serves as a crucial regularizer to evaluate the competence gap between the current policy state and the oracle target.

Consequently, this formulation is not an algebraic identity, but a regularized *Surrogate Gradient* that decouples data-driven direction from policy-driven step-size. Standard SFT applies the distributional shift but simplifies the policy magnitude to a static value of 1, yielding:

$$\nabla_\theta \mathcal{J}_{\text{SFT}} \propto \mathbb{E}_{(q,o^*)\sim\mathcal{D}}\left[1 \cdot \nabla_\theta \log \pi_\theta(o^* \mid q)\right] \tag{8}$$

2.3 OSW-FT Gradient

Advantage-Guided SFT Magnitude. We argue that while the update *frequency* and *direction*

Algorithm 1 Online Self-Weighted Fine-Tuning

Require: Dataset $\mathcal{D}$, Model π_θ, Rollout K, Batch Size B

while not converged **do**
  $\mathcal{B} = \{(q_i, o_i^*)\}_{i=1}^{B} \sim \mathcal{D}$
  Initialize buffer $\mathcal{M} \leftarrow \emptyset$
  for each q_i in $\mathcal{B}$ **do**
    Sample K outputs $\{\hat{o}_{i,k}\} \sim \pi_\theta(\cdot|q_i)$
    Calculate rewards and success probability $\hat{p}_i$
    Compute weight $w_i = 1 - \hat{p}_i$
    Store tuple (q_i, o_i^*, w_i) into $\mathcal{M}$
  end for
  Compute weighted SFT loss on **current** π_θ:

$$\mathcal{L}(\theta) = -\frac{1}{|\mathcal{M}|} \sum_{(q,o^*,w)\in\mathcal{M}} w \cdot \log \pi_\theta(o^* \mid q)$$

  Update parameters.
end while

should be guided by the oracle o^*, the *magnitude* must respect the variance-reduced baseline. Retaining the oracle's deterministic frequency while restoring the advantage magnitude $weight$ yields the **OSW-FT** gradient:

$$\nabla_\theta \mathcal{J}_{\text{OSW}} \propto \mathbb{E}_{(q,o^*)\sim\mathcal{D}}\big[(1-\hat{p}(q)) \cdot \nabla_\theta \log \pi_\theta(o^* \mid q)\big] \tag{9}$$

where $\hat{p}(q) = \frac{1}{K}\sum_{k=1}^{K} R(\hat{o}_k)$ is an online Monte Carlo estimate of the success probability. Consequently, $(1 - \hat{p}(q))$ serves as a principled heuristic that modulates the expert trajectory update using the model's empirical advantage.

2.4 Discussion

OSW-FT preserves the original data distribution and dynamically assesses the learning value of each sample online via model rollouts, re-allocating only the gradient magnitude through the advantage weight. This ability to adaptively optimize directly on datasets with unlabeled difficulty eliminates the need for manual data sorting. Unlike rollout-optimized RFT methods such as GRPO, the gradient direction in OSW-FT is anchored by the high-quality expert demonstration o^*; only the magnitude is modulated by the model's current capability. This retains supervised optimization while incorporating an online estimate of the model's current success rate. The additional online sampling relative to standard SFT is K inference-only rollouts per query. In our experiments, $K = 2$ captures a useful advantage signal under a limited online rollout budget. Finally, OSW-FT has a gradient-level structural connection to SFT and RL. Appendix C presents this connection under policy-gradient theorem (Sutton et al., 1999; Schulman et al., 2017).

3 Theoretical Analysis

We establish two key theoretical properties of OSW-FT: robustness to finite rollout estimation and convergence guarantees with smaller gradient second moment than SFT. Full proofs are deferred to Appendix B.

3.1 Robustness to Finite Rollouts

Success probability $p_s(q)$ is estimated via K rollouts: $\hat{p}(q) = \frac{1}{K}\sum_{k=1}^{K} R(\hat{o}_k)$. Denoting $g_0(q) := \nabla_\theta \log \pi_\theta(o^*|q)$, we analyze the OSW-FT estimator $\hat{g}_K(q) = (1 - \hat{p}(q)) \cdot g_0(q)$.

Theorem 1 (Finite Rollout Properties)**.** *For a fixed query q with $p := p_s(q) \in (0, 1)$ and K i.i.d. rollouts, the OSW-FT estimator is unbiased: $\mathbb{E}_{\hat{p}}[\hat{g}_K(q)] = (1 - p) \cdot g_0(q)$.*

Remark 1 (Choice of K)**.** *The signal loss probability p^K predicts a qualitative transition at $K = 2$. At $K = 1$, the weight is binary ($\{0, 1\}$) and a query with $p = 0.5$ loses its signal with 50% probability, reducing OSW-FT to hard example selection. At $K = 2$, signal loss drops to $p^2 = 25\%$ and the intermediate weight $w = 0.5$ enables soft discrimination. For $K \geq 2$, p^K is already small and further increases yield diminishing returns.*

3.2 Convergence Analysis

A subtlety of OSW-FT is that the weight $1 - p_s(q)$ depends on the current policy π_θ, yet Algorithm 1 holds it fixed within each batch of gradient updates. To handle this, we define a surrogate objective $J^{(t)}(\theta)$ at each outer iteration t, which freezes the weights at their current values while optimizing only the log-likelihood (see Definition 1 in Appendix). Under standard smoothness and bounded-gradient assumptions (Assumption 2 in Appendix), the following guarantee holds.

Theorem 2 (Convergence Rate)**.** *Suppose OSW-FT runs for T total gradient steps with learning rate $\eta = 1/\sqrt{T}$, updating weights every S steps. Let L be the smoothness constant of $J^{(t)}$, B an upper bound on $|\log \pi_\theta(o^*|q)|$, and ϵ_p the maxi-*

mum change in $p_s(q)$ between consecutive weight updates. Then:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\Big[\|\nabla_\theta J^{(\lfloor t/S \rfloor)}(\theta_t)\|^2\Big] \leq \frac{2\Delta_0 + L\cdot\Sigma^2}{\sqrt{T}} + B\cdot\epsilon_p, \tag{10}$$

where Δ_0 is the initial optimality gap. The stochastic-gradient second moment is

$$\Sigma^2 := \mathbb{E}_{q\sim D}\Big[\Big((1-p_s(q))^2 + \frac{p_s(q)(1-p_s(q))}{K}\Big)\|g_0(q)\|^2\Big], \tag{11}$$

which satisfies $\Sigma^2 < \Sigma^2_{\mathrm{SFT}} := \mathbb{E}_{q\sim D}[\|g_0(q)\|^2]$ for any $K \geq 1$, whenever the model has nonzero success probability on a positive-measure set of queries.

The first term vanishes at rate $O(1/\sqrt{T})$; the second term $B\cdot\epsilon_p$ reflects the cost of holding weights fixed between updates, which is small under typical learning rates. The strict inequality $\Sigma^2 < \Sigma^2_{\mathrm{SFT}}$ implies that OSW-FT converges with smaller second moment than SFT under identical conditions.

4 Experiments

4.1 Experimental Setup

We evaluate our method across model scales with the Qwen3 series (0.6B, 1.7B, and 4B) (Yang et al., 2025) to comprehensively study the interaction between model capacity and our dynamic weighting mechanism. We compare OSW-FT against standard SFT and GRPO (Shao et al., 2024).

Training Dataset. To establish a rigorous testbed for reasoning capabilities, we curated a high-quality training set of 10k multiple-choice questions (MCQs) sampled from AM-Qwen3-Distilled (Tian et al., 2025) — a NuminaMATH dataset (LI et al., 2024) distilled via Qwen3-235B. To ensure the provision of high-quality oracle trajectories (o^*) required for both SFT and OSW-FT, we manually filter and clean the dataset. Specifically, we rigorously enforce a single-choice answer format and constrain the reasoning trajectories to a maximum length of 4k tokens with the vast majority around 2k tokens. This length distribution ensures that even small-capacity base models can learn from substantial reasoning steps efficiently.

Experiment Details. All models undergo full-parameter fine-tuning. We employ the AdamW optimizer (Loshchilov and Hutter, 2019) with a constant learning rate schedule, alongside `DeepSpeed ZeRO3` and `bfloat16` precision across all model scales. To minimize generation overhead, we integrate `vLLM` (Kwon et al., 2023) for both training rollouts and evaluation. Its high-throughput capabilities enable us to maintain a large training rollout batch size (128), making the online cost of OSW-FT highly efficient.

The maximum sequence length is set to 5k tokens, with a maximum completion length of 4k tokens allocated for online rollouts. For OSW-FT and GRPO, we use Math-Verify (Huggingface, 2025) for binary accuracy reward function.

- **SFT:** We fine-tune the models for 1 epoch with a learning rate of 1×10^{-5}.
- **OSW-FT (Ours):** Following the same 1-epoch schedule and 1×10^{-5} learning rate. For each query, we generate $K=2$ trajectories at a temperature of $T=1.0$ to estimate the online success probability $\hat{p}(q)$.
- **GRPO:** For each query, we generate $K=8$ trajectories (group size of $G=8$) at a temperature of $T=1.0$. We adopt 1×10^{-6} learning rate, aligning with established community best practices for stable RL training. To mitigate variance from trajectory lengths, we employ token-level policy gradient loss (DAPO, (Yu et al., 2025)). We set KL penalty coefficient $\beta=0$ (disabling KL) and apply clipping bound of $[0.8, 1.28]$.

Evaluation Protocol. To thoroughly assess verifiable reasoning capabilities, we report performance on several challenging benchmarks, including advanced mathematical reasoning tasks such as AMC (AMC, 2023), AIME (AIME, 2025), and MATH-500 (Lightman et al., 2023), as well as the expert-level general question answering GPQA-Diamond (Rein et al., 2024). We report **Pass@1/Pass@16** for competition benchmarks (AMC, AIME). We use Pass@16 as an empirical indicator of exploration: under the same 16-sample budget, a higher Pass@16 means that the model can discover a correct solution on more queries through sampling. Additionally, for benchmarks comprising hundreds of items (MATH-500 and GPQA-Diamond), we report **Pass@1** as an

Table 1: **Main Results.** Comparison across different post-training paradigms. We report **Pass@1 / Pass@16** for competition benchmarks (AIME, AMC) to highlight both stability and exploration potential. For MATH-500, GPQA-Diamond, we report **Pass@1**. **Bold** indicates the best performance. † indicates the base model shows lower performance due to limited instruction-following ability, which recovers after fine-tuning.

Model	AIME 24	AIME 25	AMC 22	AMC 23	AMC 24	MATH-500	GPQA-D
Qwen3-0.6B-Base	**00.83** / 06.67	00.21 / 03.34	**11.05 / 41.86**	11.82 / 54.35	04.58 / 26.67	22.00	14.14
+ GRPO	**00.83 / 10.00**	00.00 / 00.00	08.14 / 25.58	**16.30** / 43.48	**08.61 / 31.11**	**36.20**	**26.77**
+ SFT	00.00 / 00.00	00.21 / 03.34	09.01 / 32.56	11.55 / 50.00	03.89 / 24.44	24.20	15.66
+ OSW-FT	**00.83** / 06.67	**01.04 / 10.00**	08.72 / **41.86**	13.04 / **56.52**	05.14 / 28.89	24.20	20.20
Qwen3-1.7B-Base	03.75 / 20.00	**03.96** / 20.00	**23.84 / 58.14**	28.67 / **65.22**	14.17 / **48.89**	56.40	25.76
+ GRPO	**06.25** / 23.33	02.08 / 13.33	20.20 / 48.84	**34.10 / 65.22**	13.06 / 42.22	**59.40**	**28.79**
+ SFT	02.92 / 20.00	03.12 / 10.00	18.31 / 55.81	26.90 / **65.22**	13.75 / **48.89**	52.20	17.17
+ OSW-FT	05.42 / **26.67**	**03.96 / 23.34**	22.24 / 55.81	27.45 / **65.22**	**17.78 / 48.89**	57.40	26.27
Qwen3-4B-Base	09.58 / 30.00	06.88 / 23.34	36.63 / 69.77	39.27 / **78.26**	29.31 / 60.00	69.60	10.10 †
+ GRPO	05.63 / 20.00	05.20 / 23.33	39.53 / 58.14	34.92 / 67.39	22.92 / 53.33	71.40	30.32
+ SFT	12.29 / 26.67	15.42 / 26.67	40.70 / 69.77	44.57 / 73.91	**35.00 / 75.56**	76.40	27.27
+ OSW-FT	**13.33 / 36.67**	**16.25 / 36.67**	**41.42 / 74.42**	**45.11 / 78.26**	**35.00 / 75.56**	**79.00**	**32.32**

auxiliary test to evaluate deterministic problem-solving accuracy on relatively simpler or out-of-distribution (OOD) tasks. We apply sampling with Top_p=0.95, $Top_k = 20$, and a sampling temperature of $T = 0.6$, allowing a maximum generation length of 8k tokens.

4.2 Main Results

Table 1 presents the main comparative results. Overall, OSW-FT improves over standard SFT across the evaluated small-to-medium model scales. Its comparison with GRPO varies across models and metrics, reflecting a trade-off rather than a general performance ordering between expert-anchored training and RFT.

Enhancing Exploration in Small Language Models. For the 0.6B, 1.7B, and 4B models, OSW-FT consistently outperforms or matches standard SFT on Pass@16. For instance, on the 4B model, OSW-FT achieves a Pass@16 of 36.67% on both AIME 24 and AIME 25, compared with 26.67% for SFT. From these Pass@16 results, we observe that OSW-FT encourages more exploration under the same sampling budget: it discovers correct solutions for a larger fraction of queries across repeated samples. Relative to GRPO, OSW-FT performs favorably on several competition-math metrics, while GRPO remains stronger in some other settings, particularly on Pass@1 for the smaller models.

The Capacity-Data Mismatch Phenomenon. The optimization dynamics of OSW-FT further explain its scale-dependent behavior. As illustrated in Figure 1, the advantage weight $(1 - \hat{p})$ acts

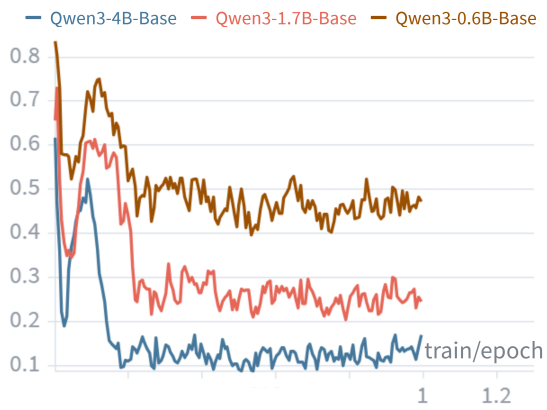

Figure 1: OSW-FT weight $(1 - \hat{p})$ stratifies by model scale, validating the model-data matching hypothesis.

as an implicit curriculum, decaying as the model masters the dataset. Crucially, these trajectories cleanly stratify by model capacity: the highly capable 4B model masters the dataset rapidly, causing its weights to plummet toward zero, whereas the 0.6B model maintains a sustained learning signal.

4.3 Performance under Limited Rollout Budgets

As established in Remark 1 (Section 3), our theoretical analysis of the signal loss probability predicts a critical qualitative transition at $K = 2$. At this minimal threshold, the advantage weight shifts from binary hard-example mining to fine-grained soft discrimination. Guided by this theoretical insight, a central computational consideration is empirically verifying this optimal trade-off against the overhead introduced by the online rollouts. Because these rollouts strictly serve to estimate the baseline for variance reduction rather than to

Table 2: **Ablation on the number of rollouts (K).** We report **Pass@16** for AIME 24, AIME 25 and **Pass@1** for MATH-500, GPQA-D on Qwen3-0.6B-Base. OSW-FT maintains robust performance at $K = 2$, while GRPO is more sensitive to the rollout count on the AIME benchmarks in this setting.

Rollouts	GRPO				OSW-FT			
	AIME 24	AIME 25	MATH-500	GPQA-D	AIME 24	AIME 25	MATH-500	GPQA-D
$K = 1$	–	–	–	–	**06.67**	06.67	22.80	15.15
$K = 2$	00.00	00.00	29.00	22.72	**06.67**	**10.00**	24.20	20.20
$K = 4$	06.67	00.00	**37.00**	26.27	**06.67**	**10.00**	24.00	**20.71**
$K = 8$	**10.00**	00.00	36.20	**26.77**	**06.67**	**10.00**	**24.60**	**20.71**

Table 3: **Ablation on alternative weighting strategies ($K = 4$).** We report **Pass@16** for AIME 24, AIME 25 and **Pass@1** for MATH-500, GPQA-D on Qwen3-0.6B-Base. OSW-FT consistently outperforms alternative weighting heuristics, validating the necessity of smooth online success-rate anchoring.

Strategy	AIME 24	AIME 25	MATH-500	GPQA-D
Hard-thresholded	**06.67**	06.67	23.80	15.15
Random	00.00	00.00	21.40	15.17
OSW-FT	**06.67**	**10.00**	**24.00**	**20.71**

discover the correct reasoning path from scratch, minimal generation overhead should theoretically suffice. To validate this, we conduct an ablation study on the Qwen3-0.6B-Base model, varying $K \in \{1, 2, 4, 8\}$. The results are shown in Table 2.

OSW-FT. As shown in Table 2, setting $K = 1$ acts as a highly noisy, binarized estimator and yields the lowest performance across all metrics (e.g., 15.15% on GPQA-D). However, simply increasing the rollouts to $K = 2$ produces a sharp performance improvement, jumping to 20.20% on GPQA-D and matching or exceeding $K = 1$ on all other benchmarks. Critically, the performance plateaus for $K > 2$. Increasing the rollouts to $K = 4$ or $K = 8$ yields only negligible marginal benefits. Consequently, a coarse estimate via $K = 2$ captures the necessary advantage signal without demanding excessive generation overhead.

Contrast with GRPO under Limited Rollout Budgets. To fully contextualize this computational efficiency, we contrast OSW-FT with the GRPO baseline under matched rollout budgets. Table 2 suggests a potential limitation of pure RL: while GRPO can optimize shorter-horizon tasks (MATH, GPQA) at minimal compute ($K = 2$), it obtains zero scores on the most challenging reasoning benchmarks, scoring **0.00** on both AIME 24 and AIME 25. This outcome may stem from RL's reliance on model-generated exploration. On highly complex tasks like AIME, the probability of finding a correct long-horizon reasoning path with only 2 random attempts can be low. Consequently, the advantage signal for these hard queries can become sparse, providing the model with limited learning signal. To alleviate this exploration bottleneck and achieve non-zero AIME performance, GRPO can benefit from larger group sizes (e.g., $K = 8$).

Advantage of GT Anchoring. This contrast suggests a potential advantage of OSW-FT for capacity-constrained models. By deterministically anchoring the gradient direction to the ground-truth oracle trajectory (o^*), OSW-FT provides a stable, expert-anchored optimization path across the evaluated tasks. Thus, it can mitigate the exploration bottleneck that can affect pure RL on hard queries, showing consistent performance across the evaluated benchmarks while operating at a limited computational budget ($K = 2$).

4.4 Ablation on Weighting Strategies

To thoroughly investigate whether the performance gains of OSW-FT stem from the mathematically derived online success-rate weighting rather than naive data filtering or generic optimization artifacts, we conduct a rigorous ablation study on Qwen3-0.6B-Base under rollout budget of $K = 4$. We implement two critical control baselines:

- **Hard-thresholded Weighting:** A discrete control scheme where a query receives a binary weight $w \in \{0, 1\}$. Specifically, $w = 0$ if all K rollouts succeed (indicating the query is fully mastered), and $w = 1$ otherwise. This mimics hard rejection sampling or hard mistake-driven learning.

- **Random Weighting:** A control scheme that decouples the loss weights from the model's actual performance. It assigns random weights drawn from a uniform distribution, calibrated to match the exact empirical mean weight of OSW-FT across the dataset, thereby testing if our method simply acts as an implicit learning rate decay.

(a) GRPO learning rate of 1×10^{-6}

(b) GRPO learning rate of 1×10^{-5}

Figure 2: GRPO training reward under two learning rates. In our setup, the reward collapses toward zero at 1×10^{-5} across the evaluated model scales, whereas 1×10^{-6} produces the stable runs used in the main comparison.

The empirical results are summarized in Table 3. We observe a severe performance degradation when utilizing alternative heuristics. In contrast, OSW-FT scales down gradient updates as capability increases, supporting the benefit of content-aware weighting under a limited online rollout budget.

4.5 On the Optimization Stability of GRPO

Learning Rate Sensitivity. We initially trained the GRPO baseline using the 1×10^{-5} learning rate employed by SFT and OSW-FT. In our setup, this learning rate causes the training reward to collapse toward zero across the evaluated model scales, as shown in Figure 2b. Once all sampled trajectories receive the same reward, the relative-advantage signal vanishes and optimization stalls.

We therefore use 1×10^{-6} for the reported GRPO baseline. This lower learning rate produces the stable training dynamics shown in Figure 2a.

Late-Stage Instability and Checkpoint Selection. We also observe late-stage degradation in some GRPO runs, particularly for the smaller-capacity models. To report a stable GRPO baseline, we evaluate the checkpoint at which the reward reaches a plateau, before subsequent degradation, rather than using the final checkpoint unconditionally.

5 Related Work

Offline Reweighting and Regularization. A growing body of work views standard SFT through an offline Reinforcement Learning lens, aiming to mitigate over-optimization without incurring the computational cost of online generation. One line of research focuses on reweighting (e.g., SoftDedup (He et al., 2024), iw-SFT (Qin and Springenberg, 2025), DFT (Wu et al., 2026)). Another line introduces structural regularization, such as ASFT (Zhu et al., 2026), which apply KL-divergence penalties against a static reference model to prevent catastrophic forgetting and stylistic overfitting. While these approaches improve upon uniform SFT, they operate entirely offline. OSW-FT bypasses these static heuristics by operating online, directly tracking the model's continuously evolving capabilities via real-time rollouts.

Online Rollout-Driven Alignment. To incorporate real-time generative feedback, recent paradigms increasingly rely on online rollouts (Schulman et al., 2017; Shao et al., 2024; Yu et al., 2025; Zheng et al., 2025; Gao et al., 2025; Wang et al., 2025; Chen et al., 2025). STaR (Zelikman et al., 2022) filters and trains exclusively on self-generated rationales that yield correct final answers. RL or hybrid-RL methods like GRPO (Shao et al., 2024) and CHORD (Zhang et al., 2026) are inherently bottlenecked by massive compute overhead. Conversely, rollout-augmented SFT methods attempt to simplify this optimization but introduce new flaws. OSFT (Li et al., 2025) fine-tunes on model-generated trajectories without filtering by correctness, and lacks explicit mechanisms for accuracy-based gradient scaling. OTR (Ming et al., 2025) guides SFT using a localized policy gradient. However, its one-token lookahead provides only myopic, token-level credit signals, lacking explicit trajectory-level credit assignment for long-horizon reasoning. OSW-FT resolves these bottlenecks by unifying trajectory evaluation with deterministic optimization.

Data Selection and Curriculum Learning. Recent findings highlight that alignment depends heavily on data quality rather than sheer quantity (Zhou et al., 2023; Li et al., 2024). To maximize data efficiency, traditional curriculum learning (Bengio et al., 2009) presents training examples in a manually predefined order of increasing difficulty. This static paradigm has largely evolved into Automatic Curriculum Learning (Narvekar et al., 2020; Portelas et al., 2020), where intermediate tasks are dynamically selected based on the agent's real-time competence. More recently, curriculum paradigms have been adapted for LLMs (Shi et al., 2025), alongside advanced data selection methods like FisherSFT (Deb et al., 2025). However, these LLM-specific approaches typically rely on static heuristics that struggle to accurately track the model's continuously evolving capabilities during training. OSW-FT seamlessly shifts the optimization budget toward the model's true capability frontier, entirely eliminating the need for offline filtering or manual data sorting.

6 Conclusion

We introduced **Online Self-Weighted Fine-Tuning (OSW-FT)**, a simple method that augments supervised fine-tuning with online, trajectory-level weighting for verifiable reasoning. By estimating the model's current success rate with a small number of inference-only rollouts, OSW-FT adjusts how strongly each expert trajectory should update the model while keeping the optimization direction anchored to expert demonstrations. Empirically, OSW-FT consistently improves over SFT on small-to-medium models and offers a favorable performance–rollout trade-off when the available online rollout budget is limited.

Limitations

OSW-FT currently assumes a reliable binary verifier and high-quality expert trajectories. Its success-rate weight is designed for binary-verifiable reasoning and has not been validated for continuous rewards or open-ended generation. Although GPQA-Diamond broadens the evaluation beyond mathematics, it remains a binary-verifiable multiple-choice task and therefore does not establish applicability to non-binary settings.

Our experiments also focus on Qwen3 model family and primarily mathematical reasoning tasks. Extending OSW-FT to code generation and other reward formulations remains future work.

References

AIME. 2025. AIME problems and solutions 2025. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

AMC. 2023. AMC problems and solutions.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In ICML.

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. 2025. Minimax-m1: Scaling test-time compute efficiently with lightning attention. Arxiv.

Rohan Deb, Kiran Koshy Thekumparampil, Kousha Kalantari, Gaurush Hiranandani, Shoham Sabach, and Branislav Kveton. 2025. FisherSFT: Data-efficient supervised fine-tuning of language models using information gain. In ICML.

Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. 2025. Soft adaptive policy optimization. Arxiv.

Saeed Ghadimi and Guanghui Lan. 2013. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. SIAM Journal on Optimization.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. Arxiv.

Nan He, Weichen Xiong, Hanwen Liu, Yi Liao, Lei Ding, Kai Zhang, Guohua Tang, Xiao Han, and Yang Wei. 2024. Softdedup: an efficient data reweighting method for speeding up language model pre-training. In ACL.

Huggingface. 2025. Math-Verify. https://github.com/huggingface/Math-Verify.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles.

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. 2024. Numinamath.

Mengqi Li, Lei Zhao, Anthony Man-Cho So, Ruoyu Sun, and Xiao Li. 2025. Online sft for llm reasoning: Surprising effectiveness of self-tuning without rewards. Arxiv.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. In NAACL.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In ICLR.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In ICLR.

Rui Ming, Haoyuan Wu, Shoubo Hu, Zhuolun He, and Bei Yu. 2025. One-token rollout: Guiding supervised fine-tuning of llms with policy gradient. Arxiv.

Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E Taylor, and Peter Stone. 2020. Curriculum learning for reinforcement learning domains: A framework and survey. JMLR.

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

Rémy Portelas, Cédric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves Oudeyer. 2020. Automatic curriculum learning for deep rl: A short survey. Arxiv.

Chongli Qin and Jost Tobias Springenberg. 2025. Supervised fine tuning on curated data is reinforcement learning (and can be improved). Arxiv.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In COLM.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. Arxiv.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. Arxiv.

Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. 2025. Efficient reinforcement finetuning via adaptive curriculum learning. Arxiv.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In NeurIPS.

Xiaoyu Tian, Yunjie Ji, Haotian Wang, Shuaiting Chen, Sitong Zhao, Yiping Peng, Han Zhao, and Xiangang Li. 2025. Not all correct answers are equal: Why your distillation source matters. Arxiv.

Jiakang Wang, Runze Liu, Lei Lin, Wenping Hu, Xiu Li, Fuzheng Zhang, Guorui Zhou, and Kun Gai. 2025. Aspo: Asymmetric importance sampling policy optimization. Arxiv.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. Arxiv.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, and 17 others. 2025. DAPO: An open-source LLM reinforcement learning system at scale. In NeurIPS.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. In NeurIPS.

Wenhao Zhang, Yuexiang Xie, Yuchang Sun, Yanxi Chen, Guoyin Wang, Yaliang Li, Bolin Ding, and Jingren Zhou. 2026. On-policy RL meets off-policy experts: Harmonizing supervised fine-tuning and reinforcement learning via dynamic weighting. In ICLR.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. 2025. Group sequence policy optimization. Arxiv.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. In NeurIPS.

He Zhu, Junyou Su, Peng Lai, Ren Ma, Wenjia Zhang, Linyi Yang, and Guanhua Chen. 2026. Anchored supervised fine-tuning. In ICLR.

A Ethics Statement

We acknowledge the use of Large Language Models (LLMs) during the preparation of this manuscript. Specifically, LLMs were utilized solely as auxiliary tools for minor writing assistance (such as grammar checking, typo correction, and phrasing refinement) and for debugging code implementations. We have reviewed all LLM-assisted output and take full responsibility for the accuracy, originality, and integrity of the work presented in this paper.

B Theoretical Proofs

Throughout, we use $p := p_s(q) = \mathbb{E}_{o\sim\pi_\theta}[R(o)|q]$ and $g_0(q) := \nabla_\theta \log \pi_\theta(o^*|q)$ for a fixed query q satisfying Assumption 1.

Assumption 1 (Binary Verifiable Reward)**.** *For each query q, the reward is binary: $R(o) \in \{0, 1\}$, and the success probability $p_s(q) := \mathbb{E}_{o\sim\pi_\theta}[R(o)|q] \in (0, 1)$ is well-defined.*

B.1 Finite Rollout Analysis (Proof of Theorem 1)

Proof. Let $\hat{p}(q) = \frac{1}{K}\sum_{k=1}^{K} R(\hat{o}_k)$ where $\hat{o}_k \sim \pi_\theta(\cdot|q)$ are i.i.d. rollouts. Since $R(\hat{o}_k) \sim \text{Bernoulli}(p)$, we have $\mathbb{E}[\hat{p}] = p$ and $\text{Var}[\hat{p}] = p(1-p)/K$.

Unbiasedness. Since $g_0(q)$ is deterministic for a fixed query q:

$$\mathbb{E}_{\hat{p}}[\hat{g}_K(q)] = \mathbb{E}[1-\hat{p}(q)] \cdot g_0(q) = (1-p) \cdot g_0(q). \tag{12}$$

Signal loss. The event $w(q) = 0$ occurs iff $\hat{p}(q) = 1$, i.e., all K rollouts are correct. By independence:

$$P(w(q) = 0) = P(\hat{p}(q) = 1) = p^K. \tag{13}$$

□

The following results provide additional quantitative characterization of the estimator under finite rollouts.

Remark 2 (Total second moment)**.** *Using the identity $\mathbb{E}[X^2] = (\mathbb{E}[X])^2 + \text{Var}[X]$ with $X = 1 - \hat{p}$:*

$$\mathbb{E}_{\hat{p}}[\|\hat{g}_K(q)\|^2] = \left[(1-p)^2 + \frac{p(1-p)}{K}\right] \|g_0(q)\|^2. \tag{14}$$

This decomposes into a signal term $(1-p)^2\|g_0\|^2$ and an estimation noise term $\frac{p(1-p)}{K}\|g_0\|^2$ that vanishes as $K \to \infty$. This expression is used in the convergence analysis (Theorem 2).

Remark 3 (Estimation-induced MSE)**.** *The MSE due to weight estimation can be isolated as:*

$$\mathbb{E}_{\hat{p}}\left[\|\hat{g}_K(q) - (1-p)g_0(q)\|^2\right] = \frac{p(1-p)}{K}\|g_0(q)\|^2, \tag{15}$$

which follows from $\hat{g}_K - (1-p)g_0 = (p - \hat{p})g_0$ and $\mathbb{E}[(p-\hat{p})^2] = \text{Var}[\hat{p}] = p(1-p)/K$.

B.2 Convergence Analysis (Proof of Theorem 2)

Definition 1 (OSW-FT Surrogate Objective)**.** *At outer iteration t with parameters θ_t, define:*

$$J^{(t)}(\theta) = \mathbb{E}_{q\sim\mathcal{D}}\left[\underbrace{(1 - p_{\theta_t}(q))}_{\textit{fixed weight}} \cdot \log \pi_\theta(o^* \mid q)\right], \tag{16}$$

where the weight is computed from π_{θ_t} and held fixed during inner-loop updates.

Remark 4 (Necessity of stop-gradient)**.** *Differentiating the naive objective $\mathbb{E}_q[(1 - p_\theta(q)) \log \pi_\theta(o^*|q)]$ through the weight would introduce a $\nabla_\theta p_\theta(q)$ term that the OSW-FT estimator $(1 - \hat{p})\, g_0$ does not capture. The surrogate $J^{(t)}$ makes explicit that $(1 - \hat{p})\, g_0$ is an unbiased gradient of $J^{(t)}(\theta)$ at $\theta = \theta_t$, which is the quantity optimized in practice. This matches Algorithm 1, where rollouts and weights are computed once per batch before gradient updates.*

Assumption 2 (Regularity Conditions)**.** *(i)* ***L-smoothness****: For each fixed weight configuration, $J^{(t)}(\theta)$ is L-smooth in θ.*

(ii) ***Bounded second moment****: $\mathbb{E}_{q,\hat{p}}[\|(1 - \hat{p}(q))\, g_0(q)\|^2] \le M$ for all θ.*

(iii) ***Weight stability****: $|p_{\theta_{t+1}}(q) - p_{\theta_t}(q)| \le \epsilon_p$ for all q between consecutive outer iterations.*

(iv) ***Bounded log-likelihood****: $|\log \pi_\theta(o^*|q)| \le B$ for all θ, q.*

Proof of Theorem 2. The proof proceeds in four steps.

Step 1: Inner-loop convergence. Within each outer iteration t, the weights are fixed, and by the unbiasedness established in Theorem 1, the stochastic gradient $(1 - \hat{p}(q))\, g_0(q)$ is an unbiased estimator of $\nabla_\theta J^{(t)}(\theta_t)$. Applying the standard SGD convergence result for L-smooth non-convex objectives (Ghadimi and Lan, 2013) over the S inner steps:

$$\frac{1}{S}\sum_{s=1}^{S}\mathbb{E}[\|\nabla J^{(t)}(\theta_{t,s})\|^2] \leq \frac{2(J^{(t)}(\theta^*)-J^{(t)}(\theta_{t,0}))}{S\eta}+L\eta\,\Sigma^2, \tag{17}$$

where

$$\Sigma^2 := \mathbb{E}_{q\sim\mathcal{D}}\left[(1-p(q))^2 + \frac{p(q)(1-p(q))}{K}\right]\|g_0(q)\|^2 \tag{18}$$

follows from the total second moment expression in Remark 2 and linearity of expectation over q.

Step 2: Cross-iteration mismatch. When switching from outer iteration t to $t+1$, the surrogate objective changes. The mismatch is bounded by:

$$\begin{aligned}&|J^{(t+1)}(\theta)-J^{(t)}(\theta)|\\ &=\left|\mathbb{E}_q\left[(p_{\theta_t}(q)-p_{\theta_{t+1}}(q))\log\pi_\theta(o^*|q)\right]\right|\\ &\leq \mathbb{E}_q\left[|p_{\theta_t}(q)-p_{\theta_{t+1}}(q)|\cdot|\log\pi_\theta(o^*|q)|\right]\\ &\leq \epsilon_p\cdot B,\end{aligned}$$

using Assumption 2(iii) and (iv).

Step 3: Accumulating the switching cost. Since there are $M = T/S$ outer-iteration transitions, the mismatch accumulates as

$$\left|\sum_{m=1}^{T/S}\left[J^{(m)}(\theta)-J^{(m+1)}(\theta)\right]\right| \leq \frac{T}{S}\epsilon_p B, \tag{19}$$

which, after dividing by $\eta T = \sqrt{T}$, contributes a term of order $(\epsilon_p B/S)\sqrt{T}$ to the averaged bound. This term is bounded by the single-transition mismatch $\epsilon_p B$ whenever $T \leq S^2$, i.e. over training horizons commensurate with the chosen update interval S:

$$\frac{\epsilon_p B}{S}\sqrt{T} \leq \epsilon_p B. \tag{20}$$

Step 4: Telescoping. Summing (17) over all T/S outer iterations and accounting for the mismatch (19) at each transition (with the switching cost bounded as in Step 3), with $\eta = 1/\sqrt{T}$:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla_\theta J^{(\lfloor t/S\rfloor)}(\theta_t)\|^2\right] \leq \frac{2\Delta_0 + L\cdot\Sigma^2}{\sqrt{T}} + B\cdot\epsilon_p. \tag{21}$$

□

Corollary 1 ($\Sigma^2 < \Sigma^2_{\text{SFT}}$)**.** *The SFT gradient second moment is $\Sigma^2_{\text{SFT}} = \mathbb{E}_q[\|g_0(q)\|^2]$. We show that $\Sigma^2 < \Sigma^2_{\text{SFT}}$ for any $K \geq 1$.*

Decompose the difference:

$$\begin{aligned}&\Sigma^2_{\text{SFT}} - \Sigma^2\\ &= \mathbb{E}_q\left[\|g_0\|^2\right]\\ &- \mathbb{E}_q\left[\left((1-p)^2 + \frac{p(1-p)}{K}\right)\|g_0\|^2\right]\\ &= \mathbb{E}_q\left[\left(p(2-p) - \frac{p(1-p)}{K}\right)\|g_0\|^2\right].\end{aligned}$$

We verify the integrand is strictly positive for $p \in (0,1)$ and $K \geq 1$:

$$\begin{aligned}&p(2-p) - \frac{p(1-p)}{K} = p\left[(2-p) - \frac{1-p}{K}\right]\\ &\geq p[(2-p)-(1-p)] = p > 0,\end{aligned} \tag{22}$$

where the inequality uses $K \geq 1$. Therefore, whenever $P(p(q) > 0) > 0$, the expectation in (22) is strictly positive, giving $\Sigma^2 < \Sigma^2_{\text{SFT}}$.

C Unifying SFT and RL via General Policy Gradient

To reveal the structural connection between Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL), we unify them under the **General Policy Gradient** framework. In RL, the gradient update for a model π_θ takes the standard form:

$$\nabla_\theta J = \mathbb{E}\left[\underbrace{(R(\tau)-b(q))}_{\text{Advantage / Weight}}\cdot\nabla_\theta\log\pi_\theta(\tau\mid q)\right] \tag{23}$$

We view SFT as **maximizing an objective function** $J_{\text{SFT}} = -\mathcal{L}_{\text{SFT}}$. Under this lens, SFT restricts the trajectory entirely to the deterministic ground-truth oracle ($\tau = o^*$, where the reward $R(o^*) \equiv 1$) and implicitly operates with a **zero baseline** ($b(q) \equiv 0$). Consequently, the SFT gradient collapses into a static, uniform update:

$$\nabla_\theta J_{\text{SFT}} \propto \mathbb{E}\left[\underbrace{(1-0)}_{\text{Static Weight}\equiv 1}\cdot\nabla_\theta\log\pi_\theta(o^*\mid q)\right] \tag{24}$$

The Missing Baseline Problem. While RL methods (such as GRPO) actively employ dynamic baselines (e.g., the group mean) to reduce variance and scale updates based on the policy's evolving capability, SFT's implicit zero baseline treats all ground-truth trajectories as having identical, maximal advantage. As a consequence, SFT assigns the exact same gradient coefficient to trivial, already-mastered samples ($\pi_\theta(o^* \mid q) \approx 1$) as it does to completely unsolved ones. This leads to inefficient gradient allocation and unnecessary variance injection during training. Notably, this inefficiency arises not from noisy supervision, but from a systematic mathematical mismatch between the static gradient estimator and the policy's true competence.

By formally deriving the variance-reduced baseline under the constraint of deterministic ground-truth trajectories, we naturally arrive at the formulation of **Online Self-Weighted Fine-Tuning (OSW-FT)**. As established in Section 2, assigning this optimal baseline to the model's current success probability ($b(q) = p_s(q)$) perfectly restores the dynamic advantage missing from standard SFT, ensuring optimal gradient reallocation while preserving the optimization stability inherent to supervised learning.