GUIDED BY THE HINT: BALANCING CAPABILITY ACQUISITION AND RETENTION IN OFFLINE AGENT FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Supervised fine-tuning (SFT) on offline agent trajectories is the standard approach for training specialized tool-using agents, but forcing models to imitate reasoning and actions token by token may harm other capabilities (e.g., general reasoning, tool calling, code generation) of the base model. In this work, we focus on studying *how SFT objectives balance the trade-off between acquiring new capabilities while preserving existing ones better on agent data?* By comparing several baselines, standard SFT improves the target benchmark while lowering several non-target benchmark scores; meanwhile, simply constraining distributional drift using KL penalty or limiting the update magnitude did not avoid this regression trend. Motivated by recent token-wise adaptive learning objectives, this work proposed **Privilege-Guided SFT (PG-SFT)** to leverage turn-level information gain of agent trajectories as indicators to adjust supervision strength. PG-SFT yields a more favorable observed trade-off on the evaluated benchmarks, substantially reducing distributional drift and broad capability degradation at the cost of slight degradation on target-task performance. Our findings suggest that balancing the acquisition–retention trade-off depends not only on whether the model is anchored to its base behavior, but also on *where* and *how* strongly supervision should depart from that behavior.

1 INTRODUCTION

Supervised fine-tuning (SFT) has been the standard approach for adapting large language models (LLMs) to follow instructions and perform specialized tasks (Ouyang et al., 2022; Touvron et al., 2023). For tool-using agents, offline execution trajectories provide supervision on reasoning and actions, allowing models to acquire task-specific behaviors without repeatedly interacting with expensive environments during training (Yao et al., 2023; Chen et al., 2024; Zeng et al., 2023).

However, token-level imitation treats every demonstrated assistant token as a target, regardless of whether it represents a task-relevant decision or an incidental realization of the trajectory. Fine-tuning can thus improve the target workflow while changing capabilities acquired before adaptation, a tension related to catastrophic forgetting (Kirkpatrick et al., 2017; Li & Hoiem, 2016; Kemker et al., 2018). Regressions in knowledge, reasoning, instruction following, and safety-related behavior have also been observed in LLM post-training (Luo et al., 2023; Qi et al., 2024; Wang et al., 2026; Xu et al., 2026; Lin et al., 2025; Huan et al., 2025; Wang et al., 2024; Biderman et al., 2024). Prior approaches reduce model–data mismatch through self-distillation (Yang et al., 2024) or constrain changes from the base model through reference-based objectives (Zhu et al., 2026a; Xu et al., 2026), but whether such control suffices to preserve capabilities remains undemonstrated; moreover, they focus on reasoning and instruction-following tasks, leaving capability retention in agent fine-tuning largely unexplored.

In this work, we study the following question: **How does supervision-objective design affect the empirical trade-off between target adaptation and non-target benchmark performance?**

Intuitively, SFT may disrupt existing capabilities less if the student stays close to its base distribution—by anchoring predictions to it or limiting successive update sizes. Yet, a model that changes too little may retain its old behavior without acquiring the target behavior. We examine this tension by holding the offline trajectories fixed and varying only the supervision objective: standard SFT provides the full-imitation reference; SFT with a Base-model KL penalty tests direct anchoring

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.

to the base distribution; and proximal SFT (PSFT) tests a constraint on updates relative to a lagged policy (Zhu et al., 2026b). Together, these methods ask whether conservative learning suffices for balancing capability acquisition and retention.

Unfortunately, extensive experiments on Qwen3 (Qwen Team, 2025) and Qwen3.5 (Qwen Team, 2026) across four benchmarks show that better fitting of the training data can coincide with larger distributional deviation from the base model, yet simply reducing this deviation with a fixed KL constraint does not necessarily produce better downstream capability retention. Trust-region baseline PSFT keeps the model closest to the base distribution, but acquires the least of the target behavior.

Recently, token-wise supervision adjustment DFT (Wu et al., 2026) and InfoSFT (Sabbaghi et al., 2026) yielded good performance in reasoning and math benchmarks. Although our preliminary analyses reveal DFT's incompatibility with agent SFT, their insights on adaptive supervision motivate us to introduce **Privilege-Guided Supervised Fine-Tuning (PG-SFT)**. PG-SFT allocates supervision turn by turn, matching the granularity at which agent decisions appear, and uses "next actions" in trajectories as training-time privileged information to measure the "knowledge gain" imposed by the hint, adjusting the supervision intensity of that turn accordingly. On Qwen3.5-4B, PG-SFT matches SFT on the selected target SWE-bench subset (59/89, pass@3) while retaining stronger performance on other non-target benchmarks. On Qwen3-4B-Thinking, PG-SFT gives up limited target-task performance in exchange for stronger retention on non-target tasks. These results suggest that the key of balancing the trade-off is not only regularizing supervision toward the base model, but deciding *where and how strongly* the model should depart from it.

Our contributions are summarized as follows:

- **An empirical study of capability acquisition and retention in offline agent fine-tuning.** We formulate post-training as a joint learning–retention problem and evaluate adaptation not only through target-task performance, but also through target fitting, distributional drift from the base model, and retained non-target capabilities.
- **PG-SFT as an adaptive probe of the learning–retention trade-off.** We introduce Privilege-Guided SFT, which uses privileged conditioning to estimate the intensity of moving away from the base distribution. PG-SFT provides a concrete mechanism for testing whether adaptive, example-dependent learning intensity can achieve a more favorable balance between acquiring specialized behavior and retaining existing capabilities.
- **A controlled comparison of different learning pressures.** Using the same offline trajectories, we compare ordinary SFT, SFT with a fixed Base-KL constraint, a clipped-ratio trust-region objective PSFT, and an adaptive supervision strategy PG-SFT. The comparison shows that fitting the target distribution, remaining close to the base distribution, and preserving downstream capabilities are distinct properties; optimizing one does not automatically guarantee the others.

The remainder of this paper is organized as follows: Section 2 reviews related work; Section 3 presents the compared objectives and PG-SFT; Section 4 reports the acquisition–retention analyses; Section 5 analyzes the turn-level coefficient and discusses the scope of our study.

2 Related Work

2.1 Capability retention in post-training

Fine-tuning enables language models to acquire specialized behaviors, but adaptation can also modify capabilities that are not directly optimized Lin et al. (2025); Huan et al. (2025); Wang et al. (2024): continual fine-tuning studies document forgetting of previously acquired knowledge and tasks Luo et al. (2023); Biderman et al. (2024), and alignment-oriented investigations show that task-specific adaptation may affect safety-related behaviors Qi et al. (2024). Existing mitigations either constrain deviation from the initial model through reference-based objectives or update rules Yang et al. (2024); Zhu et al. (2026a); Xu et al. (2026); Zhu et al. (2026b); Li et al. (2018), or preserve the original parameters through additional components Wang et al. (2026); Shenfeld et al. (2025).

2.2 Model-aware supervision and adaptive learning pressure

Several observations suggest that supervision should be calibrated to the learner: data fitting the learner is more effective Zhang et al. (2025), self-distillation reduces model–data mismatch by generating supervision closer to the learner distribution Yang et al. (2024), and smaller models struggle to absorb reasoning behaviors from substantially stronger teachers Li et al. (2025). Some work curates the data by itself to improve data-model alignment, including quality filtering Chen et al. (2023); Liu et al. (2024b), difficulty ranking Li et al. (2024a); Liu et al. (2024a), and data rewriting Li et al. (2024b), though such gains do not always scale to large pools Xia et al. (2025). Some recent studies proposed to adjust supervision strength non-uniformly to better match the data-model relationship: DFT scales token-level updates by the model's confidence on demonstrations Wu et al. (2026), IDFT accounts for contextual distribution differences Zhang et al. (2026), anchored SFT combines adaptive weighting with reference-model regularization Zhu et al. (2026a), CADFT estimates example-level compatibility Zhou et al. (2026), and TALR reweights per-token losses to reduce interference with general capabilities Lin et al. (2025). PSFT also adapts the contribution of supervised tokens, but its signal derives from the magnitude of previous model updates Zhu et al. (2026b).

2.3 Privileged information as a training-time signal

Learning using privileged information studies settings where additional information is available during training but unavailable during inference Vapnik & Izmailov (2015), and generalized distillation shows that such training-only signals improve prediction when they inform the target mapping Lopez-Paz et al. (2016). Privileged information is not universally beneficial, however: stronger privileged features do not necessarily yield better student models, motivating selective use of privileged signals Yang et al. (2022); Aslam et al. (2023); recent analyses further find that privileged teachers can bias solutions Harne et al. (2026), suppress self-correction Kaur et al. (2026), or yield unstable gains Shrestha & Tessier (2026), and that more privileged information is not automatically better Zhao et al. (2026b). In language models, privileged signals have been explored through context distillation and self-distillation mechanisms Snell et al. (2022); Agarwal et al. (2024); Zhao et al. (2026a); Penaloza et al. (2026).

3 Trade-off Balancing: A Supervision Objective Perspective

This section investigates the acquisition–retention tension through a controlled comparison of training objectives. All objectives are trained on identical data with identical supervised tokens, so that only the objective varies. We develop the comparison along two lines. The ***conservative*** line restrains the overall fitting strength with a single uniform rule, asking whether more conservative learning alone can balance acquisition against retention. The ***adaptive*** line instead varies supervision strength across decisions; our first attempt in this direction, DFT (Wu et al., 2026), underperforms even the untrained base model. Inspired by some preliminary results, we propose PG-SFT, a turn-level adaptive objective that allocates supervision with training-time privileged information.

3.1 Preliminaries and Metrics

Let p_0 be a frozen base model, p_θ the fine-tuned student, and $\mathcal{D}$ an offline dataset of agent trajectories. User messages and tool feedback are excluded from the loss. At assistant turn k, t is the position of the token in the turn, y_t is the token, and x_t is the trajectory context, including preceding turns and assistant tokens at the current turn; V is the vocabulary. Under teacher forcing, we define

$$q_{0,t}(v) = p_0(v \mid x_t), \qquad p_{\theta,t}(v) = p_\theta(v \mid x_t), \qquad v \in V. \tag{1}$$

Two teacher-forced statistics recur throughout our analysis: the average negative log-likelihood (NLL) of the demonstrated tokens, which measures how strongly a model fits the demonstrations, and the average forward KL divergence from the base model (*Base KL*), which measures how far its predictions move away from p_0 on the same contexts. Formally, for a model with per-position distributions p_t,

$$\mathrm{NLL}(p) = -\frac{1}{N}\sum_t \log p_t(y_t), \qquad \mathrm{KL}_{\text{base}}(p) = \frac{1}{N}\sum_t D_{\mathrm{KL}}(q_{0,t} \| p_t)\,, \tag{2}$$

where N is the number of supervised assistant tokens.

3.2 Conservative Supervision Objectives

Full fitting: SFT. Standard supervised fine-tuning minimizes

$$\mathcal{L}_{\text{SFT}}(\theta) = -\frac{1}{N}\sum_{t} \log p_{\theta,t}(y_t), \tag{3}$$

whose one-hot target places all probability mass on the demonstrated token. SFT supplies the full-fitting endpoint of our comparisons.

Uniformly reduced fitting: Base-KL. A natural way to retain more of the base distribution is to add a forward KL penalty to the same supervised loss:

$$\mathcal{L}_{\text{SFT+KL}}(\theta) = \mathcal{L}_{\text{SFT}}(\theta) + \frac{\lambda}{N}\sum_{t} D_{\text{KL}}(q_{0,t} \| p_{\theta,t}), \qquad \lambda \geq 0. \tag{4}$$

Up to a positive scale and terms independent of θ, this is cross-entropy against

$$r_t^{\text{SFT+KL}}(v) = \frac{\lambda}{1+\lambda} q_{0,t}(v) + \frac{1}{1+\lambda}\mathbb{1}[v = y_t]. \tag{5}$$

That is, the target interpolates between the base distribution $q_{0,t}$ and the one-hot target at the demonstrated token. We name the weight on the one-hot component the supervision coefficient α; Base-KL fixes $\alpha_{\text{KL}} = 1/(1+\lambda)$ for every turn. Sweeping λ then tests how much of the acquisition–retention behavior can be explained by reducing supervision uniformly.

Limited updates: proximal SFT. A third strategy leaves the target untouched and constrains the update instead. We consider the clipped-ratio objective of Zhu et al. (2026b), which constrains each update relative to a lagged student policy $p_{\theta_{\text{old}}}$. With a constant positive advantage and clipping radius $\epsilon > 0$, its surrogate is maximized as

$$\mathcal{J}_{\text{PSFT}}(\theta) = \mathbb{E}_{(x_t,y_t)\sim\mathcal{D}}\left[\min(\rho_t, \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon))\right], \qquad \rho_t = \frac{p_{\theta,t}(y_t)}{p_{\theta_{\text{old}},t}(y_t)}. \tag{6}$$

This rule limits probability increases on the demonstrated token relative to the lagged reference, but defines neither an intermediate target nor an anchor of the output distribution to $q_{0,t}$; it tests whether controlling updates relative to the recent policy improves retention. We therefore compare PSFT empirically, without treating its clipping radius as an α value.

Together, these objectives form the conservative family of our comparison: they control how strongly the student fits the demonstrations—either by interpolating the target between the base distribution and the demonstrated token, or by limiting each update relative to the recent policy. However, none of them conditions the target on the data–model relationship. Whether *adaptive* allocation of supervision matters beyond its overall strength is the question we take up next.

3.3 Adaptive Supervision Objectives: Preliminary Attempts on Token-wise Adjustment

Token probability is a misaligned signal for agent SFT. Our first attempt in this family is DFT (Wu et al., 2026): it assigns supervision strength token by token according to the model's probability, i.e., higher-probability tokens receive more supervision. However, evaluation of DFT on Qwen3.5 backbone only resolves **40** SWE-bench (Jimenez et al., 2024) questions, substantially lower than **54** resolutions for Base [1]. Figure 1 illustrates two potential explanations: i) the tokens receiving the strongest supervision are usually dispensable words or fixed protocol segments, which carry little information about a turn's reasoning logic or action decision; ii) the low-probability tokens are a mixture of stylistic preferences and key agent behaviors, so down-weighting them by probability also suppresses the supervision that matters.

[1] DFT uses the same training and evaluation setup in Section 4.

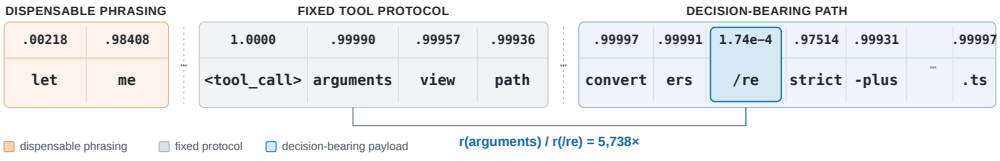
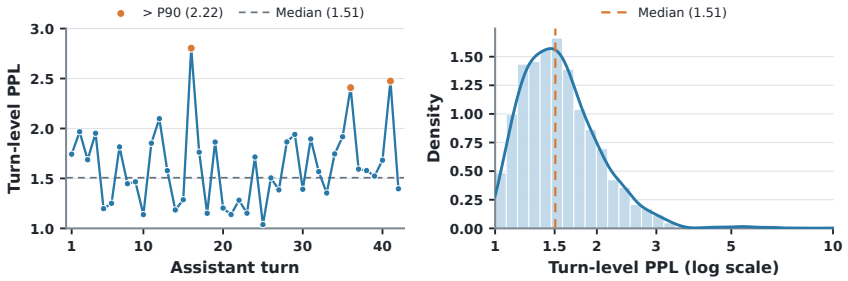


Figure 1: A typical case of DFT's supervision allocation. The value above each token is its target-token probability $p_t = p_{\theta,t}(y_t)$. DFT weights each token's cross-entropy by the stop-gradient of this value, $w_t = \mathrm{sg}[p_t]$; the per-token gradient-norm ratio then equals this weight, $\|\nabla_\theta \ell_t^{\mathrm{DFT}}\|_2 / \|\nabla_\theta \ell_t^{\mathrm{SFT}}\|_2 = w_t$.

Agent turns carry distinguishable decision signals. In an agent trajectory, a turn couples reasoning with the action it produces, making an agent turn the smallest unit in which a complete decision appears. Moreover, a token-local score may miss how a decision is expressed, whereas a signal defined over the whole turn can better reflect the inherent characteristics of the demonstrated decision. Figure 2 exhibits this potential using average token perplexity (PPL) as the probing metric: within a single trajectory, per-turn perplexity spreads around a median of 1.51 up to a P90 of 2.22 (Figure 2a); across the training data, PPL is unimodal and right-skewed with a long tail (Figure 2b), showing distinguishable characteristics between turns.



(a) within a training trajectory.

(b) across the training data.

Figure 2: Turn-level perplexity distribution of the base model on the training data.

However, a turn-level statistic by itself does not say which quantity should drive allocation. Raw perplexity, in particular, attributes a turn's surprise to everything the model does not expect, including stylistic expressions, and does not separate the part that follows from the turn's action decision. To this end, we derive a signal in the next section to isolate such components.

3.4 Adaptive Supervision Objective: Privileged-Guided Supervised Fine-Tuning

Motivated by preliminary insights, we leverage the idea of "privileged context" to construct a turn-level signal with this attribution. As illustrated in Figure 3, a "hint" derived from the turn's demonstrated next action is inserted before its assistant tokens. PG-SFT then evaluates the *same frozen base model* on the same target tokens under the ordinary prefix x_t and the hint-augmented prefix x_t^+. Let $q_{+,t}(v) = p_0(v \mid x_t^+)$, $b_t = q_{0,t}(y_t)$, and $s_t = q_{+,t}(y_t)$.

Formally, for turn k with supervised token positions T_k, we define the turn's positive gain G_k, its available probability room H_k, and its learning coefficient as

$$G_k = \sum_{t \in T_k} [s_t - b_t]_+, \qquad H_k = \sum_{t \in T_k} (1 - b_t), \qquad \alpha_k = \begin{cases} G_k / H_k, & H_k > 0, \\ 0, & H_k = 0. \end{cases} \tag{7}$$

where $[z]_+ = \max(z, 0)$. Since $0 \le G_k \le H_k$, the same $\alpha_k \in [0, 1]$ applies to both reasoning and action tokens in the turn. Intuitively, α_k measures the fraction of the turn's probability room that revealing the action resolves. A turn whose content largely follows from its action scores high—the base model already supports such tokens once the decision is known, so fitting the turn mostly

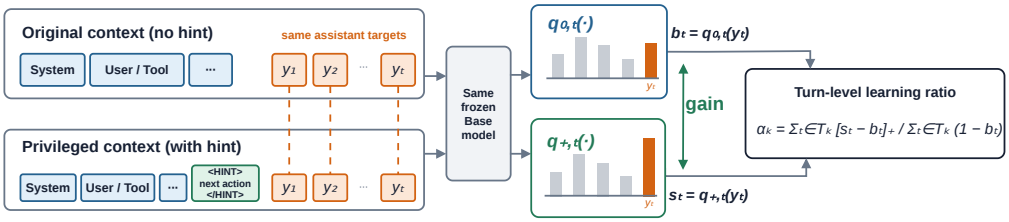


Figure 3: The same frozen base model scores aligned assistant tokens with and without an action hint. The resulting positive gain determines a shared turn-level coefficient α_k.

teaches the mapping from decision to realization. A turn that remains idiosyncratic even given the action scores low, and forcing its tokens would move the model away from the base distribution without acquiring decision-relevant behavior. Since the hint may reveal information unavailable at inference, such a coefficient is a measurement indicator of how action-determined a turn is, rather than the final supervision.

Based on this coefficient, we interpolate the target between the base distribution and the demonstrated token for every supervised token in the turn, as in Eq. 5 but with a per-turn coefficient:

$$\boxed{r_t^{\mathrm{PG}}(v) = (1-\alpha_k)q_{0,t}(v) + \alpha_k \mathbb{1}[v = y_t], \qquad t \in T_k.} \tag{8}$$

The target increases the demonstrated token's probability by $\alpha_k(1-b_t)$ while preserving the base model's relative probabilities among the other tokens; in particular, it does not distill the hint-conditioned distribution $q_{+,t}$. We train against this target with full-vocabulary forward KL (Hinton et al., 2015) with the frozen base model as teacher:

$$\mathcal{L}_{\text{PG-SFT}}(\theta) = \frac{1}{N_{\mathcal{B}}} \sum_{i\in\mathcal{B}} \sum_{k\in i} \sum_{t\in T_k} D_{\mathrm{KL}}\left(r_t^{\mathrm{PG}} \| p_{\theta,t}\right), \qquad N_{\mathcal{B}} = \sum_{i\in\mathcal{B}} \sum_{k\in i} |T_k|. \tag{9}$$

Here $\mathcal{B}$ is a batch of trajectories, i indexes a trajectory, and k indexes its assistant turns; gradients flow only through p_θ.

Specifically, to separate *where* supervision is allocated from its total target shift, we compare PG-SFT with a uniform coefficient

$$\bar{\alpha} = \frac{\sum_k G_k}{\sum_k H_k} = \frac{\sum_k \alpha_k H_k}{\sum_k H_k}, \tag{10}$$

computed across the training turns. It matches the total increase in demonstrated-token probability induced by PG-SFT, $\sum_k \alpha_k H_k$. Consequently, this uniform target coincides with Base-KL at $\lambda = (1-\bar{\alpha})/\bar{\alpha}$ for $0 < \bar{\alpha} \leq 1$.

4 Experiments

4.1 Experimental Setup

We evaluate the acquisition–retention trade-off among the supervision objectives; evaluation metrics, benchmark construction, and training hyperparameters are in Appendices C, D, and B.

Models and training data. We select two 4B-scale model families, Qwen3.5-4B Qwen Team (2026) and Qwen3-4B-Thinking-2507 Qwen Team (2025). All fine-tuning objectives use the same offline agent trajectories sampled from the open-source Nemotron-SFT-SWE-v3 dataset NVIDIA (2026); for Qwen3.5-4B, the main experiments use 1K randomly sampled OpenHands Wang et al. (2025) trajectories; for Qwen3-4B-Thinking-2507, we increase the training set to 5K to observe differentiable behavior after SFT.

Benchmarks. SWE-bench Lite Jimenez et al. (2024) share similar trajectory format as the training data and serves as the primary target-task benchmark. GPQA Diamond Rein et al. (2024), BFCL v4 Patil et al. (2025), and LiveCodeBench Jain et al. (2025) evaluate capabilities retained during

fine-tuning, from the aspects of general reasoning, multi-turn tool use, and code generation. Given the limited capability of 4B-scale backbones, some benchmarks are evaluated on their easier subsets to keep the scores informative. We report the **pass@3** metric Chen et al. (2021) to alleviate the impact of random sampling of agentic tasks, which also aligns with the mode coverage effect of SFT Gu et al. (2024).

Baselines. We evaluate five models for each backbone: **Base**, **SFT**, **SFT+Base-KL**, **PSFT**, and **PG-SFT**, our adaptive objective described in Section 3.4. The four fine-tuning objectives observe exactly the same training data and the same supervised assistant tokens. PSFT uses its published trust-region radius and optimization budget (Appendix B).

Hardware. All training and evaluation used a single server with one Intel Xeon Platinum 8480C CPU, two NVIDIA RTX PRO 6000 Blackwell GPUs, and 512 GB of host memory.

4.2 Target Adaptation and Capability Retention

Table 1 reports the absolute success counts and two aggregate statistics: **Macro Avg**, the equal-weight average of the four benchmark success rates, and **Δ vs. Base**, its change over the Base model in accuracy percentage points (pp).

		Target Adaptation	**Cross-Benchmark Capability**			**Overall**	
Backbone	**Objective**	**SWE-bench ↑** /89	**GPQA ↑** /198	**BFCL ↑** /200	**LCB ↑** /120	**Macro Avg ↑** (%)	**Δ vs. Base ↑** (pp)
Qwen3.5-4B	Base	54	151	111	60	60.61	0.00
	SFT	**59**	133	100	48	55.87	−4.74
	SFT+Base-KL	57	130	105	44	54.72	−5.89
	PSFT	53	142	**112**	**63**	59.94	−0.67
	PG-SFT	**59**	**150**	111	61	**62.10**	+1.49
Qwen3-4B-Thinking	Base	18	131	102	92	53.51	0.00
	SFT	**31**	127	48	82	47.83	−5.69
	SFT+Base-KL	10	**133**	83	87	48.10	−5.41
	PSFT	12	125	108	**88**	50.99	−2.53
	PG-SFT	27	131	**110**	85	**55.58**	+2.07

Table 1: Target adaptation and cross-benchmark capability retention. Macro Avg averages the four per-benchmark success rates with equal weight; Δ vs. Base is its change over the Base model in accuracy percentage points (pp), computed within each backbone.

Results reveal a clear tension between fitting the target workflow and retaining capabilities already present in the Base model. Standard SFT improves the target task but regresses on all three non-target benchmarks. Adding a Base-KL regularizer reduces distributional deviation during training, but does not alleviate these downstream regressions. Thus, globally encouraging the model to remain close to its initial distribution is not sufficient to preserve capabilities.

PSFT achieves one of the worst target-task results among the SFT objectives while retaining the most non-target capability among the baselines, aligning with findings that a trust-region constraint improves retention (Zhu et al., 2026b); it nevertheless fails to learn effectively from the demonstrations.

PG-SFT exhibits a different acquisition–retention profile. It resolves 59 of the 89 instances (66.3%) on Qwen3.5, the highest target-task result we measure, while staying within a few instances of Base on non-target benchmarks. On Qwen3, PG-SFT gives up some target-task performance in exchange for stronger retention on non-target tasks. This balanced profile also holds in aggregate: PG-SFT attains the highest Macro Avg on both backbones (62.10% and 55.58%) and is the only fine-tuning objective that improves over its own Base model (+1.49 pp and +2.07 pp), whereas every other objective loses capability overall (−0.67 to −5.89 pp).

Despite being trained on the same offline demonstrations, objectives can induce substantially different changes outside the target task: stronger imitation does not necessarily translate into a better

overall adaptation outcome, and a fixed Base-KL constraint does not reliably determine which existing behaviors should be preserved. To understand factors that may contribute to these differences, we next examine how each objective influences the learning process and distribution updates.

4.3 How Much Do Objectives Learn from the Data?

We evaluate the distribution alignment of each fine-tuned model with the same 120 held-out agent trajectories from the Nemotron-SFT-SWE-v3 dataset. For all supervised assistant tokens, we measure their negative log-likelihood (NLL) and the forward KL divergence $D_{\mathrm{KL}}(p_0 \| p_\theta)$ between the frozen Base distribution and the fine-tuned model, i.e., the two statistics defined in Eq. 2. Results are summarized in Table 2.

	Qwen3.5-4B					Qwen3-4B-Thinking				
Metric	**Base**	**SFT**	**SFT+ Base-KL**	**PSFT**	**PG-SFT**	**Base**	**SFT**	**SFT+ Base-KL**	**PSFT**	**PG-SFT**
NLL $\downarrow$	0.4302	0.3756	0.3767	0.4217	0.3968	0.7672	0.4903	0.5309	0.7423	0.5748
Base KL $\downarrow$	-	0.0732	0.0431	0.0026	0.0080	-	0.2454	0.1105	0.0014	0.0342

Table 2: Teacher-forced analysis on held-out offline trajectories for both backbones.

KL regularization holds the drift, but cannot retain capability. Standard SFT fits the demonstrations most aggressively and produces the largest observed distributional shift from the Base model. SFT+Base-KL nearly halves this shift ($0.0732 \rightarrow 0.0431$) at essentially the same held-out NLL, yet the reduction does not translate into better capability retention (Table 1).

Learning conservatively withholds training towards the target. PSFT fits the demonstrations least and stays closest to the Base distribution (Base KL 0.0026), yet yields the worst target-task performance and the second-best capability profile (Table 1).

PG-SFT demonstrates clearer learning signals and distribution drift. PG-SFT improves held-out NLL over the Base model while keeping Base KL at 0.0080, substantially below SFT and SFT+Base-KL: it absorbs information from the data, but more conservatively than SFT.

Interpreting these results together, if learning conservatively is sufficient to preserve capabilities, would uniformly constraining the learning intensity at the fixed level used by PG-SFT yield the same acquisition–retention trade-off?

4.4 Is Conservative Learning Sufficient?

We test this hypothesis with a **matched-intensity control**. PG-SFT assigns a turn-dependent learning intensity α_k while keeping the target on the Base-to-demonstration path of Eq. 8; weighting each turn by its supervised token count $|T_k|$, the average intensity over our Qwen3.5 training set is $\bar{\alpha}_{\text{token}} \approx 0.35$. We therefore construct a fixed objective $\mathcal{L}_{\text{Fixed}} = 0.35\,\mathcal{L}_{\text{CE}} + 0.65\,D_{\mathrm{KL}}(q_0 \| p_\theta)$, i.e., the same intensity $\alpha = 0.35$ for every supervised token. The control applies one learning intensity everywhere, whereas PG-SFT determines α_k separately for each assistant turn from the privileged signal.

As shown in Table 3, the fixed-intensity model closely matches PG-SFT's aggregate offline statistics (NLL 0.4006 vs. 0.3968; Base KL 0.0057 vs. 0.0080): the control learns approximately the same teacher-forced drift while remaining at least as close to the Base distribution.

Objective	**NLL**	**Base KL**
Fixed intensity	0.4006	0.0057
PG-SFT	0.3968	0.0080

Table 3: Offline statistics for PG-SFT v.s. the matched fixed-intensity control.

However, similar aggregate statistics do not produce the same downstream behavior. As shown in Table 4, the fixed-intensity model drops from $59/89$ to $50/89$ relative to PG-SFT on the SWE-bench subset and from $150/198$ to $140/198$ on GPQA, while BFCL stays at a similar level and LiveCodeBench even moves in the opposite direction. These divergent outcomes confirm the significance of turn-adaptive supervision. PG-SFT's turn-specific intensity concentrates updates on turns whose content is resolved by

the action hint and spares general-purpose tokens, producing selective changes that a single global coefficient cannot reproduce.

Objective	**SWE-bench** /89	**GPQA** /198	**BFCL** /200	**LCB** /120
Base	54	151	111	60
Fixed intensity	50	140	114	72
PG-SFT	59	150	111	61

Table 4: Downstream performance of PG-SFT vs. the matched fixed-intensity control.

Why turn-adaptive supervision matters? The observed PG-SFT coefficients are highly non-uniform across turns (Figure 4): the turn-level median is 0.244, the 90th percentile 0.607, and the token-weighted mean near 0.35. Combined with Tables 3 and 4, this indicates the significance of *which supervised decisions are allowed to move away from the Base distribution*.

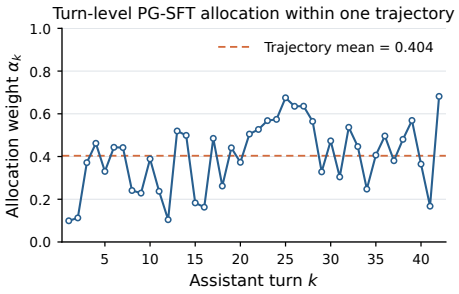


Figure 4: PG-SFT learning intensity α.

Overall, the experiments support four observations: i) stronger fitting of the training data may accompany substantial regressions in untrained capabilities; ii) a fixed Base-KL penalty reduces distributional drift but does not guarantee capability retention; iii) restricting update magnitudes, as PSFT does, keeps the model closest to the Base distribution yet commits to neither target acquisition nor non-target retention; iv) matching PG-SFT's average learning intensity does not reproduce its downstream behavior, motivating adaptive allocation of learning pressure across supervised decisions rather than a single global rule.

5 Discussion and Conclusion

Correlations between turn coefficient and turn length. α_k is an average value, and thus could strongly correlate with the number of tokens in a turn. We analyzed 5,503 assistant turns from 120 held-out trajectories to understand their relationship. While the turn length correlates with G_k and H_k individually ($r = 0.66$ and 0.75), the two dependencies largely cancel in the ratio, leaving only a weak correlation with α_k (Pearson $r = 0.225$, cluster-bootstrap 95% CI $[0.184, 0.269]$, univariate $R^2 \approx 5\%$). Therefore, despite the turn-level averaging of α_k, the coefficient reflects more on what a turn contains rather than how many tokens it has.

Scope & Limitations. Our distributional measurements are empirical, and cannot establish a causal link between drift and retention. Our evaluation covers two related 4B-scale backbones, a single agent domain, and a limited number of training runs. Therefore, we cannot promise our findings can be extended to training data and model size at any scale: although fine-tuning regressions have been documented in larger models (Qi et al., 2024; Luo et al., 2023), data-selection effects are capacity-dependent (Dang et al., 2026), so neither a pure capacity account nor a pure objective account can be excluded. Finally, privileged scoring and full-vocabulary KL add computation over ordinary SFT, although the privileged scores are computed once for a dataset and a model.

Conclusion. This work studies how supervision-objective design affects the acquisition–retention trade-off in offline agent fine-tuning. Comparing standard SFT, Base-KL, PSFT, and PG-SFT on identical trajectories shows that target fitting, base proximity, and downstream retention are distinct: SFT forgets non-target capabilities; Base-KL reduces drift without reliable retention gains; PSFT stays near the base model but learns little. PG-SFT improves the trade-off through adaptive, turn-level supervision. Balancing acquisition and retention thus depends not merely on anchoring to the base model, but on deciding where and how strongly to depart from it.

AI use statement

We used AI coding assistants to implement the various training objectives and the surrounding experimental pipeline. We also used AI tools to assist with literature searches, related-work indexing, and language editing of the manuscript. All code, citations, related-work coverage, and generated text were subsequently reviewed, verified, and corrected by the authors; the authors additionally performed manual literature searches to supplement and cross-check the related work. No scientific data were generated with AI tools.

Ethics statement

This work uses only publicly available datasets and models. We do not foresee any direct ethical concerns with the work.

Reproducibility statement

Section 3 defines the training objectives and turn-level target construction. Appendix A describes normalization and the chunked KL computation; Appendix B specifies the reported training contract; and Appendices C and D describe the evaluation sets and scoring.

References

Rishabh Agarwal et al. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations (ICLR)*, 2024.

Muhammad Haseeb Aslam, Muhammad Osama Zeeshan, Marco Pedersoli, Alessandro L. Koerich, Simon Bacon, and Eric Granger. Privileged knowledge distillation for dimensional emotion recognition in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3338–3347, 2023.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, et al. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*, 2024.

Lichang Chen et al. AlpaGasus: Training a better alpaca with fewer data. arXiv:2307.08701, 2023.

Mark Chen et al. Evaluating large language models trained on code. arXiv:2107.03374, 2021.

Zehui Chen, Kuikun Liu, Qiuchen Wang, Wenwei Zhang, Jiangning Liu, Dahua Lin, Kai Chen, and Feng Zhao. Agent-FLAN: Designing data and methods of effective agent tuning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Cuong Dang, Hoang Anh Just, and Ruoxi Jia. Capacity-dependent effects of data selection for reasoning. arXiv:2608.13721, 2026.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Sarthak Harne et al. Privileged, but biased: How PI-conditioned teachers break self-distillation. arXiv:2608.04794, 2026.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general LLM capabilities? understanding transferability of LLM reasoning. arXiv:2507.00432, 2025.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations (ICLR)*, 2025.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations (ICLR)*, 2024.

Simran Kaur et al. Rethinking on-policy self-distillation for thinking models. arXiv:2607.05184, 2026.

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

James Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. *PNAS*, 2017.

Ming Li et al. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024a.

Ming Li et al. Selective reflection-tuning: Student-selected data recycling for LLM instruction-tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024b.

Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *ICML*, 2018.

Yuetai Li et al. Small models struggle to learn from strong reasoners. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision (ECCV)*, pp. 614–629, 2016.

Jiacheng Lin, Zhongruo Wang, Kun Qian, Tian Wang, Arvind Srinivasan, Hansi Zeng, et al. SFT doesn't always hurt general capabilities: Revisiting domain-specific fine-tuning in LLMs. arXiv:2509.20758, 2025.

Liangxin Liu et al. SelectIT: Selective instruction tuning for LLMs via uncertainty-aware self-reflection. arXiv:2402.16705, 2024a.

Wei Liu et al. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *International Conference on Learning Representations (ICLR)*, 2024b.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR)*, 2016.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. arXiv:2308.08747, 2023.

NVIDIA. Nemotron-SFT-SWE-v3: Software engineering instruction tuning data, 2026. URL `https://huggingface.co/datasets/nvidia/Nemotron-SFT-SWE-v3`. Agentic trajectories from the OpenHands, SWE-agent, and mini-SWE-agent harnesses for SWE-Bench style tasks.

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

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *International Conference on Machine Learning (ICML)*, 2025.

Emiliano Penaloza et al. Privileged information distillation for language models. arXiv:2602.04942, 2026.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations (ICLR)*, 2024.

Qwen Team. Qwen3 technical report. arXiv:2505.09388, 2025. Checkpoint: `https://huggingface.co/Qwen/Qwen3-4B-Thinking-2507`.

Qwen Team. Qwen3.5: Towards native multimodal agents, 2026. URL `https://qwen.ai/blog?id=qwen3.5`. Checkpoint: `https://huggingface.co/Qwen/Qwen3.5-4B`.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. In *First Conference on Language Modeling (COLM)*, 2024.

Mahdi Sabbaghi, George J. Pappas, Adel Javanmard, and Hamed Hassani. InfoSFT: Learn more and forget less with information-aware token weighting. arXiv:2605.14967, 2026.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. RL's razor: Why online reinforcement learning forgets less. arXiv:2509.04259, 2025.

Samyak Shrestha and Alexander Tessier. Rethinking privileged information in on-policy self-distillation. arXiv:2608.18271, 2026.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. arXiv:2209.15189, 2022.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, et al. Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288, 2023.

Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: Similarity control and knowledge transfer. *Journal of Machine Learning Research*, 16(61):2023–2049, 2015.

Jiarui Wang, Xiang Shi, Jiaqi Cao, Rubin Wei, et al. MemSFT: Mitigating alignment tax with an external parametric memory. arXiv:2607.25614, 2026.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, et al. OpenHands: An open platform for AI software developers as generalist agents. In *International Conference on Learning Representations (ICLR)*, 2025.

Yihan Wang, Si Si, Daliang Li, Michal Lukasik, Felix Yu, Cho-Jui Hsieh, Inderjit S. Dhillon, and Sanjiv Kumar. Two-stage LLM fine-tuning with less specialization and more generalization. In *International Conference on Learning Representations (ICLR)*, 2024.

Yongliang Wu et al. On the generalization of SFT: A reinforcement learning perspective with reward rectification. In *International Conference on Learning Representations (ICLR)*, 2026.

Tingyu Xia et al. Rethinking data selection at scale: Random selection is almost all you need. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025.

Runze Xu, Arpit Garg, Hemanth Saratchandran, and Simon Lucey. Mask the target: A plug-and-play regularizer against LoRA forgetting. arXiv:2605.29498, 2026.

Shuo Yang, Sujay Sanghavi, Holakou Rahmanian, Jan Bakus, and S. V. N. Vishwanathan. Toward understanding privileged features distillation in learning-to-rank. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 26658–26670, 2022.

Zhaorui Yang et al. Self-distillation bridges distribution gap in language model fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Aohan Zeng et al. Agenttuning: Enabling generalized agent abilities for llms. *arXiv preprint arXiv:2310.12823*, 2023.

Dylan Zhang, Qirun Dai, and Hao Peng. The best instruction-tuning data are those that fit. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Miaosen Zhang et al. Towards on-policy SFT: Distribution discriminant theory and its applications in LLM training. arXiv:2602.12222, 2026.

Siyan Zhao et al. Self-distilled reasoner: On-policy self-distillation for large language models. arXiv:2601.18734, 2026a.

Xuyang Zhao et al. Is more privileged information better? from solution traces to problem-solving structure in self-distilled reasoning. arXiv:2608.01589, 2026b.

Yucheng Zhou et al. Compatibility-aware dynamic fine-tuning for large language models. arXiv:2606.11206, 2026.

He Zhu et al. Anchored supervised fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2026a.

Wenhong Zhu, Ruobing Xie, Rui Wang, Xingwu Sun, Di Wang, and Pengfei Liu. Proximal supervised fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2026b.

A Implementation details

Distributed normalization. The objective in Eq. 9 is defined once per optimizer update but computed per microbatch. With R data-parallel ranks and K_k microbatches actually executed per rank in update k, the implementation injects the global divisor

$$d_k = \frac{N_k}{RK_k}, \qquad \ell_{r,b} = \frac{1}{d_k} \sum_{(i,t)\in\mathcal{B}_{r,b}} D_{\mathrm{KL}}(r_t \| p_{\theta,t}) \,, \tag{11}$$

so that averaging over ranks and microbatches recovers the objective exactly,

$$\frac{1}{R}\sum_{r=1}^{R} \frac{1}{K_k}\sum_{b=1}^{K_k} \nabla_\theta \, \ell_{r,b} = \nabla_\theta \, \mathcal{L}_{\text{PG-SFT}}. \tag{12}$$

Here N_k is the number of supervised assistant tokens in update k (cf. $N_{\mathcal{B}}$ in Eq. 9 for a full batch), and $\mathcal{B}_{r,b}$ is the set of token positions in microbatch b on rank r. Trailing updates with fewer than K_k microbatches divide by the number actually executed.

Exact KL without materializing the batch. Both logarithms in Eq. 9 require the full vocabulary at every supervised position, which a standard forward pass would materialize as a tensor of size $N_k \times |V|$. The implementation instead receives the student hidden states H together with the targets r_t built from the frozen base lane, and projects the hidden states in token-row chunks C,

$$Z_C = H_C W_{\text{out}}^\top, \qquad p_C = \mathrm{softmax}(Z_C), \tag{13}$$

accumulating the exact KL chunk by chunk, so the logit workspace scales with $O(|C||V|)$ rather than $O(N_k|V|)$ while the value and gradient are those of the unchunked objective. The custom backward for the student path is

$$\frac{\partial \ell}{\partial H_C} = \frac{1}{d_k}\,(p_C - r_C)\, W_{\text{out}}, \tag{14}$$

the gradient of Eq. 9 composed with the output projection.

Cost. Relative to SFT, a PG-SFT update additionally runs the frozen base model over the ordinary and privileged contexts, constructs a full-vocabulary target at every supervised position, and retains the intermediate states that the chunked backward needs. The clipped-ratio objective instead requires the demonstrated-token probabilities of the reference policy that defines its ratio, in addition to the student forward pass. We therefore do not claim equal compute for the compared objectives, which are matched on data and supervised tokens only (Section 4). Diagnostics such as negative log-likelihood, perplexity, and entropy on the demonstrated tokens are logged for analysis and are not part of any objective.

B Training contract

Training data. The offline trajectories are sampled from the open-source Nemotron-SFT-SWE-v3 dataset NVIDIA (2026) and follow the OpenHands Wang et al. (2025) agent format, which matches the trajectory format of SWE-bench. The Qwen3.5-4B experiments use 1,000 randomly sampled trajectories; the Qwen3-4B-Thinking-2507 experiments use 4,992 trajectories, because differences between objectives were not measurable at the smaller size for that backbone.

Table 5 lists the settings shared by all compared objectives on the Qwen3.5-4B backbone and what each objective adds; only the supervision objective and its own hyperparameters differ. Table 6 reports the corresponding contract for the Qwen3-4B-Thinking-2507 backbone.

Shared setting	**Value**
Backbone	Qwen3.5-4B
Training trajectories	1,000 offline agent trajectories
Supervised region	complete assistant generation mask
Holdout panel	120 fixed trajectories
Seed	42
Precision	BF16
Adaptation	LoRA (Hu et al., 2022) rank 32, $\alpha = 64$, dropout 0 on all linear layers except `lm_head`
Optimization	AdamW, learning rate 5×10^{-5}, 5 warmup steps, gradient clip 1.0
Batch and schedule	global batch 32; cosine decay over a 64-step horizon, 32 steps executed
Checkpoint	fixed final-step checkpoint (LoRA-merged weights)
Objective	**Objective-specific settings**
SFT	cross-entropy on the demonstrated tokens (Eq. 3)
SFT+Base-KL	SFT plus forward $\mathrm{KL}(p_0 \| p_\theta)$ with $\lambda = 0.1$ (Eq. 4)
PG-SFT	frozen base lanes, temperature 1, turn-level α_k of Eq. 7
PSFT	clipped surrogate of Eq. 6 with $\epsilon = 0.28$; 3 epochs (96 optimizer steps) at learning rate 10^{-6}
Fixed intensity	$0.35\,\mathcal{L}_{\mathrm{CE}} + 0.65\,D_{\mathrm{KL}}(q_0 \| p_\theta)$ (Section 4.4); step-32 merged checkpoint

Table 5: Training contract of the compared objectives on the Qwen3.5-4B backbone. The SFT, SFT+Base-KL, PG-SFT, and fixed-intensity arms share their adaptation and optimization configuration; the PSFT arm follows its own schedule and is therefore reported as a reference point rather than as a budget-matched comparison (Section 4.1).

The $\lambda = 0.1$ coefficient of the SFT+Base-KL arm places its fixed target (Eq. 5) at $\alpha = 1/(1+\lambda) \approx 0.91$, a much weaker reduction of supervision than the matched fixed-intensity control ($\alpha = 0.35$) of Section 4.4.

In Table 6, the SFT+Base-KL arm uses the same coefficient $\lambda = 0.1$ as on Qwen3.5-4B but trains for 32 steps instead of one epoch (156 steps); on this backbone its comparison against the SFT and PG-SFT arms is therefore not budget-matched, in the same sense as the PSFT schedule.

C Evaluation protocol

Benchmark sets. GPQA Diamond Rein et al. (2024) uses all 198 questions in the frozen set. BFCL v4 Patil et al. (2025) uses a fixed 200-instance subset of the multi-turn categories (50 per category)

Shared setting	**Value**
Backbone	Qwen3-4B-Thinking-2507
Training trajectories	4,992 offline agent trajectories
Supervised region	complete assistant generation mask
Holdout panel	120 fixed trajectories
Seed	42
Precision	BF16
Adaptation	LoRA rank 32, $\alpha = 64$, dropout 0 on all linear layers except `lm_head`
Optimization	AdamW, learning rate 5×10^{-5}, 5 warmup steps, gradient clip 1.0
Batch	global batch 32
Checkpoint	fixed final-step checkpoint (LoRA-merged weights)
Objective	**Objective-specific settings**
SFT	cross-entropy on the demonstrated tokens (Eq. 3); 156 steps (one epoch), cosine decay
SFT+Base-KL	SFT plus forward $\mathrm{KL}(p_0 \| p_\theta)$ with $\lambda = 0.1$ (Eq. 4); 32 steps, cosine decay
PG-SFT	frozen base lanes, temperature 1, turn-level α_k of Eq. 7; 156 steps (one epoch), cosine decay
PSFT	clipped surrogate of Eq. 6 with $\epsilon = 0.28$; 3 epochs (96 optimizer steps) at learning rate 10^{-6}

Table 6: Training contract of the compared objectives on the Qwen3-4B-Thinking-2507 backbone. Adaptation, optimization, seed, and data processing match Table 5; only the training-set size and the per-objective optimization budgets differ. In particular, the SFT+Base-KL arm executes 32 steps rather than a full epoch, so it is not budget-matched to the SFT and PG-SFT arms on this backbone.

scored by the official `bfcl-eval` harness; this subset is a development split of the benchmark rather than its official test split. LiveCodeBench Jain et al. (2025) uses a frozen 120-instance stratified subset of release v6. SWE-bench Jimenez et al. (2024) uses the 89-instance set of Appendix D with the official SWE-bench harness verification.

Metric. The target-task column reports **pass@3**: an instance is counted as resolved when the harness resolves at least one of its attempts. The attempt schedule of each subset of the 89 instances is given in Appendix D.

Generation budget. SWE-bench rollouts use temperature 1, at most 8192 new tokens per round, at most 150 rounds, and a context of 131072 tokens. GPQA is generated at temperature 0.6 with top-p 0.95, top-k 20, and at most 32768 new tokens in the first pass; instances whose first output is truncated at that budget are re-run with a 65536-token budget. BFCL is generated at temperature 0 with seed 42, at most 8192 new tokens per request, and a context of 262144 tokens.

Scoring. GPQA is graded by extracting the final answer under a fixed answer rule: an example counts as correct when the correct option is identifiable from the response, and examples without an identifiable final answer count as incorrect. SWE-bench, BFCL, and LiveCodeBench use their task-specific execution-based evaluators.

D SWE-bench evaluation set

The target-task results in Table 1 are evaluated on a frozen set of 89 SWE-bench Lite instances. We construct this set to improve discriminability: on a uniformly random sample of Lite, the two backbones often achieve similar performance, providing limited signal for comparing training objectives. The final set combines a random subset with a difficulty-screened subset where the two backbones exhibit clearer separation.

Official Lite split. SWE-bench Lite is constructed from the SWE-bench test set through a filtering and sampling procedure that removes low-quality or ambiguous instances and selects 300 test

instances.[2] We follow the official harness and evaluation protocol; only the evaluated instance subset differs.

Random subset. The random subset is the frozen 50-instance sample used in our earlier experiments, drawn from the 300 Lite instances with `random.sample` (seed 42). It is not difficulty-screened or repository-stratified.

Difficulty-screened subset. The screened subset is constructed using external evaluation evidence without using any model evaluated in this paper. We aggregate public SWE-bench Lite evaluation records and repeated-sampling statistics to estimate instance-level solvability, remove known flaky instances, and select 50 instances with higher and more consistent resolution frequencies. The selected instances have average resolution frequencies of 62.1% and 64.0% under the two statistics, respectively. These values characterize the selected instances and should not be interpreted as model pass rates.

Union and scoring. The two subsets overlap on 11 instances, resulting in an 89-instance evaluation set after deduplication by *instance_id*. All instances use the same agent scaffold, prompt template, and output budget. For the random subset, we follow the previous evaluation protocol; for the screened subset, we additionally evaluate up to three attempts for unresolved instances. Since additional attempts are allocated conditionally based on previous outcomes, the reported pass@3 should be viewed as a comparison protocol rather than an unbiased estimate of a pre-registered three-sample evaluation. The resulting 89-instance scores are not directly comparable to SWE-bench Lite leaderboard numbers, which are reported on the full 300-instance test set.

E Supplementary materials

The submission package includes the training code for PG-SFT and all baseline objectives (SFT, SFT+Base-KL, and PSFT), together with the evaluation harness used to produce the reported benchmark results.

[2] https://github.com/SWE-bench/SWE-bench/blob/v4.1.0/swebench/collect/make_lite/README.md