CoR-OPD: Verified Local Repair and Handback for On-Policy Distillation of Coding Agents

Anonymous authors
Paper under double-blind review

Abstract

On-policy distillation (OPD) trains a student on its own trajectories, but an early mistake can derail a long-horizon agent. Teacher predictions on the resulting trajectory may be unreliable, and token-level matching along its unproductive suffix may not teach the student how to recover. We present CoR-OPD, which detects sustained teacher–student disagreement in decision tokens, returns to the last reliable action boundary, and requests a teacher bridge. With Qwen3-Coder-30B, CoR-OPD resolves 50.0% of SWE-bench Verified, compared with 45.8% for the student and 48.8% for standard OPD. With Qwen3.5-9B, it achieves 49.85% macro rubric completion on SWE-Atlas Codebase QnA, compared with 46.07% for the base student. It also transfers to multi-turn function calling and cross-language code editing. Behavioral analysis shows that bridge supervision improves evidence probing and schema repair, supporting verified local handback as a promising training strategy for long-horizon agents.

1 Introduction

Coding agents solve repository-level tasks by repeatedly inspecting files, invoking tools, editing code, and interpreting execution feedback (Liu et al., 2024; Yang et al., 2024; Wang et al., 2025). This interaction loop underlies realistic software-repair benchmarks such as SWE-bench (Jimenez et al., 2024), but it also makes trajectories easy to derail. Because each decision shapes the context for those that follow, one locally plausible mistake can move the student away from a productive solution path and possibly into a long sequence of retries or an unpromising path.

Knowledge distillation transfers a stronger model's predictions or generated sequences into a student (Hinton et al., 2015; Kim & Rush, 2016). For autoregressive models, OPD scores student-generated sequences to reduce train–test mismatch (Agarwal et al., 2024). By aligning supervision with student-visited states, OPD has become a widely used foundation for distilling reasoning and agentic capabilities, with recent work extending it through selective, turn-aware, future-aware, and teacher-relay variants (Zhong et al., 2026; Zhou et al., 2026; Jiang & Ferraro, 2026; Xu et al., 2026a).

However, OPD assumes that student-visited states remain suitable for continued supervision. This assumption becomes fragile in long-horizon agent training. Once a student trajectory has diverged, the teacher may no longer provide reliable supervision for the remaining suffix (Zhong et al., 2026; Xu et al., 2026a). The teacher must condition on a history created by decisions that its own policy may not have taken. Its predictions may depend on incidental details of the failed path, attempt a brittle salvage, or propose locally plausible actions without a coherent route back to the solution. In agentic coding, this problem is especially severe because each edit or command changes the repository and the observations that follow, allowing the context to drift progressively farther from states where teacher guidance is dependable.

This observation suggests that long-horizon agent distillation requires not only better teacher predictions, but also mechanisms to rescue the state before continuing supervision. We present CoR-OPD, a local repair-and-handback procedure for agentic OPD. When teacher–student disagreement persists on behavioral decisions, CoR-OPD rolls back to the last complete action, executes a short teacher bridge, and then returns control to the student. Intuively, agentic coding is particularly suitable for this intervention because tool calls naturally define executable boundaries and tests or build outputs provide visible evidence for local verification.

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.

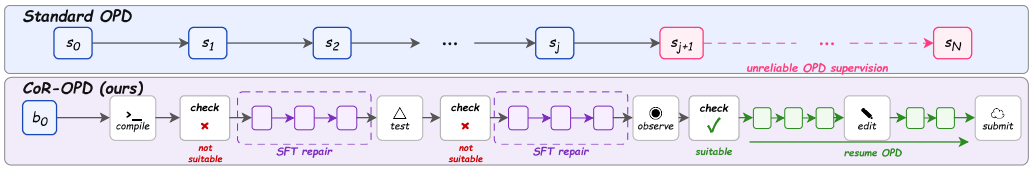

Figure 1: Method overview. **Top:** Standard OPD continues along a suffix conditioned on the student's executed states; after a consequential divergence at s_j, later supervision can remain tied to an unsuitable trajectory. **Bottom:** CoR-OPD aligns every boundary with a complete environment action. A failed check triggers a short teacher bridge trained with SFT and executed from the restored snapshot. Once a check succeeds, control returns to the student and OPD resumes.

A second challenge is that even a reliable teacher signal can be difficult to learn under standard OPD. The teacher may place most of its corrective probability mass outside the student's current support, leaving a disagreement too large for local distribution matching to absorb; recent work calls this property token teachability (Wang et al., 2026d). After a failed prefix, dense token-level targets can also produce fragmented or competing gradients instead of a coherent correction (Jiang et al., 2026a). Standard OPD evaluates the teacher only on prefixes visited by the student. It never executes the teacher's alternative at the earlier fork or exposes the student to the trajectory that follows. CoR-OPD addresses this limitation by localizing correction into short, verified repair segments rather than distilling arbitrary failed suffixes.

We compare against EPISODE TAKEOVER, a teacher-intervention baseline in which the teacher generates the rest of the episode after a detected divergence; unlike takeover, CoR-OPD preserves a student-generated suffix after correction. Figure 1 illustrates the difference between continuing distillation on potentially corrupted states and performing local recovery before resuming OPD.

Table 1 summarizes the three training policies evaluated here. All retain student-visited prefixes; episode takeover replaces the remaining rollout with teacher actions, whereas local handback lets the student practice continuing from a repaired state.

Table 1: Training policies evaluated in this work. Local handback combines environment repair with a student-generated continuation.

Policy	Student-visited prefix	Environment repair	Resumes after repair
Standard OPD	✓	–	–
Episode Takeover	✓	✓	–
CoR-OPD	✓	✓	✓

Compared with the base student, STANDARD OPD, and EPISODE TAKEOVER, CoR-OPD consistently performs best with Qwen3-Coder-30B, resolving 50.0% of SWE-bench Verified tasks and achieving the highest fine-grained rubric completion on SWE-Atlas Codebase QnA. The gains transfer beyond these settings: relative to the base student, CoR-OPD improves BFCL-v3 Multi-Turn accuracy from 28.25% to 32.00% and raises Aider Polyglot accuracy from 30.7% to 32.9% under matched decoding. With Qwen3.5-9B, CoR-OPD obtains 49.85% macro and 49.69% micro rubric completion on SWE-Atlas, compared with 46.07% and 45.56% for the base student. These evaluations cover a wide range of abilities including repository-scale recovery, codebase understanding, sequential tool use, and cross-language code editing.

Our contributions are:

- We identify a local failure mode of standard OPD in agentic environments: after environment divergence, continued suffix distillation can provide unreliable supervision even when the teacher itself is capable.
- We formulate verified local repair and handback for coding-agent OPD through decision-slot gating, environment rollback, visible validation, and selective loss.
- We compare standard OPD, episode takeover, and local handback on software repair, function calling, and codebase understanding, and measure how retained bridges affect recovery.

2 Related Work

From sequence distillation to on-policy optimization. Language-model distillation has progressed from teacher-generated targets (Kim & Rush, 2016) to student rollouts that reduce exposure bias (Agarwal et al., 2024; Gu et al., 2024). Work has since refined divergence objectives (Ko et al., 2024; 2025; Kim et al., 2025; Zhu et al., 2026b) and student-compatible supervision (Kim et al., 2024; Liu et al., 2026d;b). Subsequent studies stabilize updates through target reformulation (Jang et al., 2026), uncertainty-aware objectives (Jin et al., 2026), relaxed imitation (Ko et al., 2026), trust regions (Xing et al., 2026), corrected advantage estimation (Zhao et al., 2026b), bounded rewards (Zhao et al., 2026a), and reward extrapolation (Yang et al., 2026b).

Training dynamics and failure modes. Analyses of OPD link effective transfer to compatible teacher–student reasoning patterns (Li et al., 2026b), declining supervision quality to student-prefix drift (Liu et al., 2026e;a), and wasted optimization effort to persistent high-loss tokens (Jiang et al., 2026c); gradient-alignment diagnostics further assess whether teacher updates improve student success (Armandpour et al., 2026). Other studies examine biased or unstable loss estimators (Fu et al., 2026; Zhu et al., 2026a), length-related pathologies (Luo et al., 2026; Wang et al., 2026c), degenerate token agreement (Yu et al., 2026b), format collapse under reward extrapolation (Li et al., 2026a), reduced uncertainty expression (Kim et al., 2026), miscalibration (Zhang et al., 2026b), and cross-teacher style bias (Wu et al., 2026). Earlier analysis also shows that online data diversity can mitigate teacher hacking, where closer teacher imitation degrades alignment with the underlying target distribution (Tiapkin et al., 2025). Further studies examine teacher-origin effects on generalization (Li et al., 2026c) and gains attributable to student-side token suppression (Ding & Zhang, 2026). Parameter-space analyses characterize sparse updates and subspace locking (Shen et al., 2026; Yu et al., 2026a).

Supervision selection and training efficiency. Several works select supervision using token uncertainty and disagreement (Wang et al., 2026d; Xu et al., 2026c; Ke et al., 2026), decision-supporting evidence (Xiao et al., 2026), trajectory position (Liu et al., 2026c; Xie et al., 2026), step reliability and correctness (Zhong et al., 2026; Zheng et al., 2026; Wang et al., 2026b; Hou et al., 2026), reasoning progress (Yang et al., 2026a), student competence (Xu et al., 2026b), input-groundedness (Jiang et al., 2026b), or outcome-verified teacher guidance (Qu et al., 2026; Zhang et al., 2026c). Prompt ordering further schedules reliable supervision (Zhu et al., 2026c). Others improve efficiency through prefix training (Zhang et al., 2026a; Ziheng et al., 2026), selective truncation (Yang et al., 2026c), asynchronous generation (Rang et al., 2026; Kang et al., 2026), and update extrapolation (Cai et al., 2026). For longer interactions, improvements address temporal curricula (Wang et al., 2026a), adaptive rollout depth (Zhou et al., 2026), and future-aware alignment (Jiang & Ferraro, 2026).

3 Method

3.1 Pre-analysis: where standard OPD fails locally

Standard OPD assumes that a teacher distribution remains a useful token-level target throughout a student-generated trajectory. We test this assumption on more than 1,600 coding trajectories by evaluating the teacher on the exact student-generated sequences. To characterize where teacher and student support diverge, we define weighted high-confidence overlap (HCO). At each token position i, we construct the teacher and student support sets C_T and C_S by retaining, for each model, the smallest set of highest-probability vocabulary items whose cumulative mass reaches 0.9, capped at 16 items. We then define

$$\mathrm{HCO}_i = \frac{\sum_{v \in C_T \cap C_S} \min\{p_T(v), p_S(v)\}}{\sum_{v \in C_T \cup C_S} \max\{p_T(v), p_S(v)\}}. \tag{1}$$

The intersection rewards shared high-probability support, while the union normalizes by the probability mass represented by either policy. Nucleus truncation prevents the low-probability tail from dominating the comparison. Appendix B gives the data and metric details.

Figure 2 shows a distributional tail hidden by average overlap. Across trajectory position, mean HCO stays between 0.78 and 0.81, while 9.1–13.3% of all tokens remain below threshold. Thus, stable trajectory-level averages can conceal sharp local disagreement between teacher and student support.

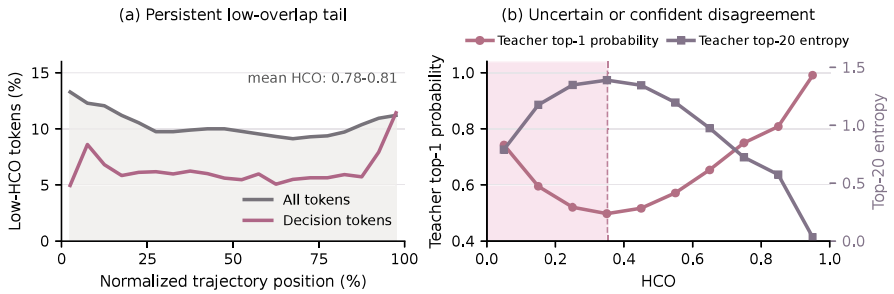

Figure 2: Local failures in standard OPD. **(a)** Mean HCO remains in a narrow range, but a low-overlap tail persists across normalized trajectory position; the late decision-token estimate uses fewer tokens. **(b)** Low overlap occurs under both diffuse uncertainty and confident disagreement. Pastel-pink shading marks the gated region.

Confidence alone does not capture teacher–student overlap. From HCO $[0.9, 1.0]$ to $[0.1, 0.2)$, teacher top-1 probability falls from 0.991 to 0.595 and truncated entropy rises from 0.032 to 1.176. Yet in the most divergent bin, HCO $[0, 0.1)$, top-1 probability returns to 0.743. HCO measures support overlap; confidence measures the strength of the teacher's preference.

We use the low-overlap tail to switch the training objective locally. Distribution matching continues in teacher-compatible regions. At a detected collapse, its loss stops and an executed, visibly verified SFT bridge returns the environment to a state from which on-policy learning can resume.

3.2 CoR-OPD overview

CoR-OPD uses a repeated control loop at action boundaries. After each complete student action, the disagreement gate checks whether teacher and student support remain aligned enough for standard OPD. If the gate triggers, the system discards the divergent suffix, restores the preceding action-aligned snapshot, and asks the teacher for a bounded bridge. The bridge contributes supervised loss only when its executed actions provide student-visible evidence of recovery. Failed attempts restart from the same snapshot and add no bridge loss. After verification, control returns to the student, which continues under its own policy. The gate is applied again at later action boundaries, allowing multiple short interventions within one trajectory.

3.3 Problem setting and training variants

Let a student policy π_θ interact with an environment through segments of reasoning and tool actions. Each executed action changes environment state s_t and produces a visible observation o_{t+1}. A frozen teacher π_T shares the student's tokenizer and tool-call schema, so teacher prompt log-probabilities can be evaluated on the exact student token sequence. We construct targets near student-visited states while replacing corrupted paths with verified local corrections.

An episode is a sequence $\tau = (s_0, x_0, a_0, o_1, \ldots, s_T, x_T, a_T)$, where x_t denotes the textual context available to the policy and a_t is either a reasoning continuation or a tool action. For an executed action, the environment applies a transition $s_{t+1} = \mathcal{E}(s_t, a_t)$ and returns o_{t+1}. The transition may be irreversible from text alone: once a_t edits a file, later observations come from the edited repository. We store snapshots at action boundaries so that rollback can restore the full training state (x_t, s_t).

Suppose teacher and student behavior first diverge inside segment t. A useful local target must preserve three invariants. The retained prefix must end at a complete action boundary; the teacher action must be executed from the corresponding snapshot; and the continuation shown to the student must contain only information observable from that snapshot. These invariants prevent the packed sequence from combining a textual prefix from one branch with an environment observation from another. They also make the handback state replayable during data inspection.

We study three data policies. Standard OPD lets the student finish a rollout and applies distillation on the fixed trajectory. Episode Takeover invokes a disagreement gate and gives the teacher control from the first repair event to the end of the episode. The full CoR-OPD policy invokes the

teacher only for a local bridge and then returns the repaired state to the student. This comparison isolates the distinction between observing the student's own errors, replacing the remainder of an episode, and learning to continue after a bounded repair.

3.4 Segment construction and action boundaries

We divide a rollout at complete interaction boundaries. Segment lengths vary because each segment contains the reasoning that selects an action, the structured tool call, and the resulting visible observation. This unit aligns the language-model sequence with the environment transition and provides a natural rollback point. The system can restore the snapshot before the action without interpreting or inverting an incomplete edit command.

3.5 Decision-slot disagreement gate

At each segment boundary, we aggregate HCO only over behavioral decision slots, including tool names, edit subcommands, shell verbs, search patterns, and paths. A sustained drop in HCO triggers REPAIR only when it covers one of these decisions. Otherwise, the segment remains in OPD.

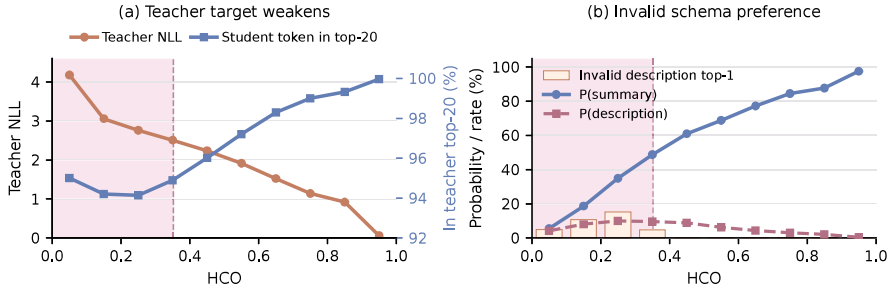

Figure 3: Diagnostics for the disagreement gate on standard-OPD trajectories. **(a)** Lower HCO coincides with higher teacher NLL on the student's realized token and lower inclusion in the teacher's top-20 support. **(b)** At legal `summary` positions, the teacher can prefer the schema-invalid `description`; all 1,069 such top-1 events occur in the pastel-pink gated region.

Figure 3 connects the gate to the loss decision. From HCO $[0.9, 1.0]$ to $[0.1, 0.2)$, teacher NLL rises from 0.062 to 3.059, while top-20 inclusion falls from 99.97% to 94.21%. In the schema probe over 35,570 legal `summary` positions, the teacher ranks the invalid `description` first at 3.01% of positions. Every such event lies in the gated region, with a mean teacher probability of 0.698 on the invalid token. We stop dense distribution matching in this region. A proposed bridge is retained only after execution and visible verification.

After evaluating a segment, the gate returns either ACCEPT or REPAIR. Under ACCEPT, the executed environment state is kept and the segment remains in the OPD objective. Under REPAIR, the loss is masked from the detected fork onward, the most recent complete action boundary is selected as rollback point j^*, and the environment snapshot stored at j^* is restored. The teacher bridge begins from this complete, well-defined repository state.

Loss masking and snapshot rollback serve different purposes. Masking removes the discarded text branch from the training objective. Rollback removes that branch's side effects from the repository before teacher intervention. Together they keep the training prefix, bridge execution state, and later handback on the same trajectory branch.

3.6 Verified local bridge and handback

From the rollback snapshot, the teacher proposes a short bridge using the repository state, failed action, and tool outputs available at that point. Each attempt is parsed and executed in the sandbox. Failed parses, rejected actions, budget warnings, and the terminal `finish` action contribute no bridge loss, and retries restart from the same snapshot.

The bridge restores a state from which the student can continue. We verify its effectiveness by running a check matched to the failure after executing the bridge: for example, schema validation for a malformed tool call, syntax validation for a code edit, a focused reproducer or targeted repository test for incorrect behavior, or a build command for an integration failure. Only a bridge that executes at least one real action and passes this check is retained and contributes supervised loss. Bridge selection uses only student-visible checks and excludes hidden tests and official evaluation labels. These requirements ensure that only executable, locally validated corrections enter training. After a successful check, the resulting snapshot and tool outputs are returned to the student, which resumes its rollout under π_θ. The resulting student-generated suffix shows whether the correction remains useful after teacher control ends.

One action boundary jointly indexes the text mask, rollback snapshot, and bridge target. This shared index keeps the retained prefix, bridge execution state, and handback state consistent and auditable.

The packed loss contains accepted student prefixes and tokens from executed actions in retained bridges. Let $\ell_{\mathrm{OPD}}(\pi_\theta, \pi_T; h_i)$ denote the token-level teacher–student distillation loss used on an accepted student prefix h_i, and let y_i^T be an executed teacher bridge token. With masks m_i^A and m_i^B for these two sources, respectively, the objective has the form

$$\mathcal{L}(\theta) = \sum_i m_i^A \ell_{\mathrm{OPD}}(\pi_\theta, \pi_T; h_i) - \lambda \sum_i m_i^B \log \pi_\theta(y_i^T \mid h_i), \tag{2}$$

where the bridge mask excludes unexecuted and terminal tokens.

The first term preserves on-policy supervision on accepted student prefixes. The second trains the executed actions that move the rollback snapshot to the verified repair state. Before packing, separate masks identify each token's source and exclude failed, rejected, unexecuted, and terminal proposals.

4 Experiments

4.1 Setup

We study two teacher–student configurations. The main experiments distill Qwen3-Coder-Next-80B-A3B (Cao et al., 2026) into Qwen3-Coder-30B-A3B-Instruct (Qwen Team, 2025). We additionally distill Qwen3.5-27B into Qwen3.5-9B to cover broader. The main evaluation compares the policies defined in Section 3.

We collect the main Qwen3-Coder training trajectories on SWE-Smith repository-repair tasks. Appendix A.2 describes the task pool, collection setup, and packed training data.

We use SWE-bench Verified (Jimenez et al., 2024; OpenAI, 2024) and SWE-Atlas Codebase QnA (Scale Labs, 2026) as primary evaluations, with BFCL-v3 (Yan, 2025) and Aider Polyglot (Aider, 2024) testing transfer to function calling and cross-language editing.

SWE-bench Verified. SWE-bench Verified contains 500 human-validated GitHub issues paired with repository snapshots and executable tests. Given a natural-language issue, the agent can inspect files, edit code, and run commands or tests before submitting a patch. We use OpenHands native function calling and official clean-sandbox grading. The primary metric is the resolved rate; we additionally report fail-to-pass (F2P) and pass-to-pass (P2P) pass rates.

SWE-Atlas Codebase QnA. SWE-Atlas Codebase QnA contains 124 questions spanning 11 repositories and 1,306 atomic factual rubrics. The agent investigates a repository through the same tool-oriented scaffold and then produces a technical answer grounded in the inspected code. Each rubric item is judged independently. Macro completion first averages rubric coverage within each question and then across questions, whereas micro completion pools all rubric items before averaging.

BFCL-v3 Multi-Turn. We evaluate on the Multi-Turn subset of BFCL-v3 (Yan, 2025), which covers basic multi-turn interactions, missing parameters, missing functions, and long-context scenarios. The model must select tools and arguments, interpret execution outputs, and carry information across successive user requests. The official evaluator checks both the resulting backend state and required function-execution paths at each turn; a conversation is correct only when all turns pass. We report Multi-Turn accuracy.

Aider Polyglot. Aider Polyglot (Aider, 2024) comprises 225 challenging Exercism programming exercises spanning C++, Go, Java, JavaScript, Python, and Rust. Each task requires the model to turn a natural-language specification into edits to source files, with correctness determined by executable unit tests. The benchmark evaluates both implementation and the ability to express applicable code changes across languages. After an unsuccessful first attempt, the model receives test feedback and can revise its solution once. We report the fraction of tasks solved within these two attempts.

4.2 END-TO-END RESULTS

Table 2: Repository-level results for Qwen3-Coder-30B and Qwen3.5-9B. All scores are percentages and higher is better. Blue and pink cells mark the highest and second-highest scores within each model and metric, respectively, with ties sharing the same color. Standard, Takeover, and COR-OPD correspond to S1, S2, and S3. F2P/P2P scores are average test pass rates on successfully applied patches for Qwen3-Coder and task-level pass rates for Qwen3.5.

Benchmark	Metric	BASE	Standard	Takeover	COR-OPD
Qwen3-Coder-30B-A3B-Instruct					
SWE-bench Verified	Resolved	45.8	48.8	47.8	**50.0**
	F2P pass	53.6	56.1	55.7	**57.7**
	P2P pass	97.1	96.0	96.8	**97.8**
SWE-Atlas QnA	Rubric macro	31.55	36.27	38.43	**39.59**
	Rubric micro	31.39	36.22	38.36	**40.12**
Qwen3.5-9B					
SWE-bench Verified	Resolved	41.3	50.2	45.4	**52.8**
	F2P pass	45.3	54.2	49.5	**56.7**
	P2P pass	67.3	72.6	72.0	**80.3**
SWE-Atlas QnA	Rubric macro	46.07	48.70	48.43	**49.85**
	Rubric micro	45.56	48.47	48.12	**49.69**

Software repair. The results in Table 2 show that COR-OPD consistently improves repository-level software repair across both teacher–student configurations. On SWE-bench Verified, COR-OPD increases the resolved rate over Standard OPD by 1.2 points for Qwen3-Coder-30B and 2.6 points for Qwen3.5-9B, while also outperforming Episode Takeover. These gains support our hypothesis that rescuing unproductive trajectories provides a more effective training signal for coding agents. Beyond recovering from failed states, returning control to the student after local repair further improves learning by allowing the policy to continue exploration under its own distribution.

COR-OPD also achieves the highest F2P and P2P scores for both students, combining more successful correction of failing tests with better preservation of existing functionality. This joint improvement is particularly valuable for coding agents since successful repository maintenance requires both resolving the reported issue and maintaining the behavior of the surrounding code.

Codebase understanding. Beyond executable repair, COR-OPD also improves repository-level understanding measured by SWE-Atlas Codebase QnA. The gains on both macro and micro rubric completion demonstrate that the learned policy is better able to inspect repositories, identify relevant evidence, and construct technically grounded answers. This result extends the impact of COR-OPD beyond patch generation, showing that the proposed training strategy improves broader codebase reasoning abilities required by autonomous coding agents.

Generalization. COR-OPD achieves the highest scores on both Aider Polyglot and BFCL-v3 Multi-Turn for both students (Figure 4), demonstrating transfer to cross-language code editing and sequential tool use. On Aider Polyglot, it improves over the base by 2.2 percentage points for Qwen3-Coder and 3.1 points for Qwen3.5; on BFCL-v3 Multi-Turn, Qwen3-Coder gains 3.75 points. These improvements extend the benefits of repository-oriented training to self-contained editing and function-calling tasks. The consistent results across both students demonstrate the generalization of COR-OPD across coding-agent workloads.

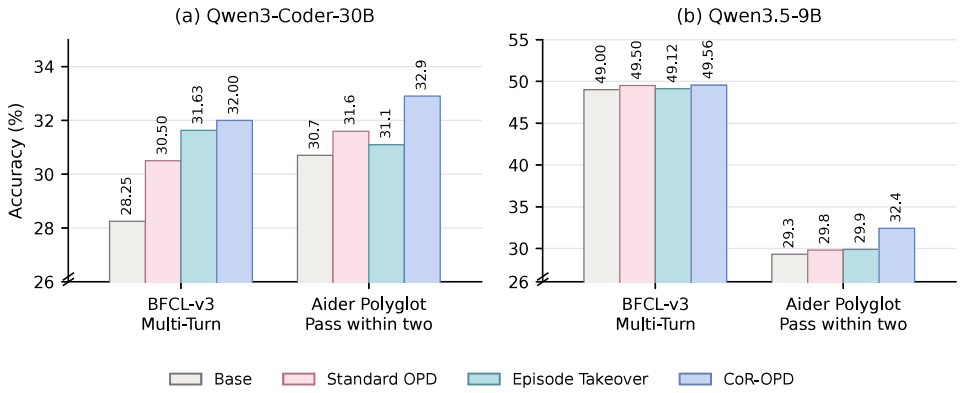

Figure 4: Results on BFCL-v3 and Aider Polyglot demonstrate that CoR-OPD generalizes to multi-turn tool use and cross-language code editing across both student models.

Overall, Local Handback consistently outperforms both Standard OPD and Episode Takeover across various tasks. This suggests that the benefit comes from accurately controlling when and how external guidance is introduced during the agent's own interaction process. Together, the results provide evidence that CoR-OPD offers a general training principle for improving long-horizon agents.

5 Analysis

5.1 What does the bridge change?

We compare two policies trained with the same disagreement gate and update budget. *Gate only (drop-bridge)* uses the gate to identify repair positions but discards the corresponding teacher-bridge examples, so no teacher-bridge tokens contribute to the loss. *Gate + bridge* uses the same gate and includes successfully executed, verified teacher-bridge actions as SFT targets.

We evaluate two recovery behaviors on controlled states taken from common coding-agent failures. *Evidence probing* asks whether the agent gathers information that can distinguish between competing explanations of a failure, such as inspecting an existing file after a create-file error, before committing to another change. *Schema repair* asks whether the agent corrects an invalid structured tool call, such as by supplying a missing required field, and resumes valid tool use. Appendix C describes the state construction and scoring criteria.

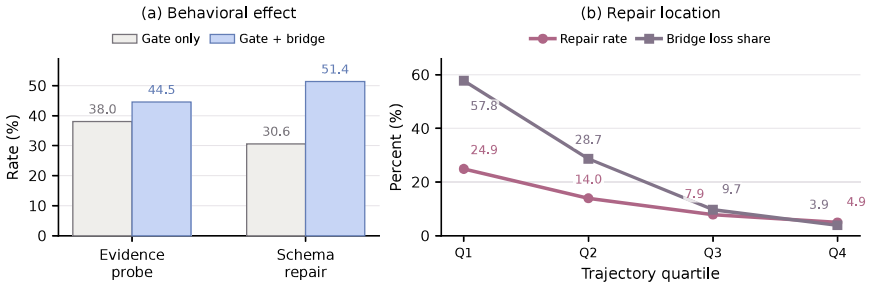

Figure 5: Bridge supervision and repair location. Left: grouped bars compare the gate-only policy with gate plus verified bridges. Right: repair rate counts only retained bridges that contribute training loss; bridge-loss share measures their allocation across four consecutive trajectory quartiles, Q1–Q4.

Adding verified bridges improves both behaviors, with the largest gain in schema repair. The executed bridge supplies an actionable correction: it restores a valid interaction and leaves the student in a state from which it can continue.

In Figure 5, retained repairs and their contribution to bridge loss concentrate in the early part of a trajectory and then decline. Most bridge supervision occurs near early forks, leaving a longer student-controlled suffix in which to practice continuing after handback.

5.2 How often should handback be reconsidered?

We ablate the granularity of the handback decision. The main policy checks for disagreement at complete segment boundaries and, when needed, executes a teacher bridge before returning control. The alternative checks after arbitrary token positions and returns control as soon as teacher and student agree again. Agreement over a short token substring cannot establish that the current branch is suitable for OPD because the action and its effect on the environment may still be incomplete. A segment boundary closes one logical interaction step: the reasoning that selects an action, the complete tool call, and the returned observation. At this point, the complete action and its observed outcome provide the context needed to evaluate disagreement and decide whether to return control. Figure 6(a) shows that boundary-aligned handback improves over the same base student more consistently across all four evaluations, with the largest advantage on SWE-Atlas QnA. These results favor completing a segment-level correction before returning control.

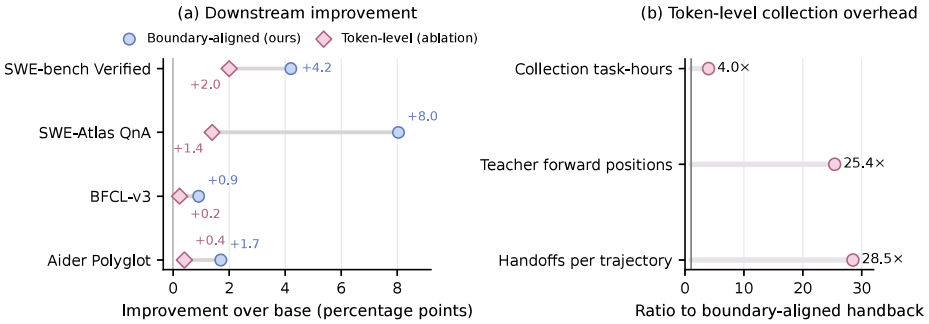

Figure 6: Handback granularity under approximately matched training-token budgets. **Left:** improvement over the base student on four benchmarks for boundary-aligned handback and the token-level ablation. **Right:** token-level collection overhead relative to boundary-aligned handback.

The intervention traces explain this gap. Token-level checking triggers 44.5 handoffs per trajectory on average, versus 1.56 for boundary-aligned handback, yet each intervention contains only 5.97 teacher tokens on average. Teacher-written tokens account for just 1.99% of its loss, compared with 40.9% for segment-level bridges. The token-level setting amounts to many short distributional corrections; it rarely gives the teacher enough tokens to execute a complete, verifiable recovery.

Token-level handback is also inefficient during data collection. Evaluating the gate at arbitrary token positions increases teacher forward positions by 25.4× and collection cost from 326 to 1,300 task-hours, with a pronounced long tail. Packed optimization reaches the matched token budget faster because nearly every generated token contributes loss, but target collection remains the bottleneck. Overall, token-level handback uses more teacher computation and control switching while producing lower downstream accuracy.

6 Conclusion and Limitations

We present COR-OPD, which repairs a diverging agent trajectory with a short verified teacher bridge and then returns control to the student. This intervention lets the student learn to continue from a corrected state while preserving student-led exploration. Extensive experiments demonstrate the method's effectiveness and generalization across coding-agent tasks, supporting local repair and handback as a practical approach to on-policy distillation.

The method requires interleaved teacher–student generation, state rollback, and executable verification, adding implementation complexity over standard OPD. Adapting these operations to static offline trajectories remains challenging. Its applicability also depends on reliable action boundaries and observable feedback; extending it to environments with weaker verification is an open direction.

AI USE STATEMENT

Generative AI tools were used to organize author-provided experimental records, assist with literature search, draft and revise prose, and generate plotting and document-build scripts. Reported experimental claims were restricted to an author-provided audited evidence summary. The authors are responsible for checking the cited literature, validating every figure and table against the underlying artifacts, reviewing AI-assisted text for originality and accuracy, and approving the final submission. The final content remains the authors' responsibility.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649.

Aider. The Aider polyglot coding benchmark. https://aider.chat/2024/12/21/polyglot.html, 2024. Accessed 2026-08-20.

Mohammadreza Armandpour, Fatih Ilhan, David Harrison, Ajay Jaiswal, Duc N.M Hoang, Fartash Faghri, Yizhe Zhang, Minsik Cho, and Mehrdad Farajtabar. Unmasking On-Policy Distillation: Where It Helps, Where It Hurts, and Why. *arXiv preprint arXiv:2605.10889*, 2026. URL https://arxiv.org/abs/2605.10889.

Yuchen Cai, Ding Cao, Liang Lin, Chunxi Luo, Xin Xu, Kai Yang, Weijie Liu, Saiyong Yang, Tianxiang Zhao, Guangzhong Sun, Guiquan Liu, and Junfeng Fang. Learning to Foresee: Unveiling the Unlocking Efficiency of On-Policy Distillation. *arXiv preprint arXiv:2605.11739*, 2026. URL https://arxiv.org/abs/2605.11739.

Ruisheng Cao, Mouxiang Chen, Jiawei Chen, Zeyu Cui, Yunlong Feng, Binyuan Hui, Yuheng Jing, Kaixin Li, Mingze Li, Junyang Lin, et al. Qwen3-Coder-Next technical report. *arXiv preprint arXiv:2603.00729*, 2026. URL https://arxiv.org/abs/2603.00729.

Yi Ding and Ruqi Zhang. Does On-Policy Distillation Really Distill? From Noisy Teacher to Self-Improvement. *arXiv preprint arXiv:2608.31046*, 2026. URL https://arxiv.org/abs/2608.31046.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting On-Policy Distillation: Empirical Failure Modes and Simple Fixes. *arXiv preprint arXiv:2603.25562*, 2026. URL https://arxiv.org/abs/2603.25562.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: On-policy distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.08543.

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

Wenjin Hou, Shangpin Peng, Weinong Wang, Zheng Ruan, Yue Zhang, Zhenglin Zhou, Mingqi Gao, Yifei Chen, Kaiqi Wang, Hongming Yang, Chengquan Zhang, Zhuotao Tian, Han Hu, Yi Yang, Fei Wu, and Hehe Fan. Uni-OPD: Unifying On-Policy Distillation with a Dual-Perspective Recipe. *arXiv preprint arXiv:2605.03677*, 2026. URL https://arxiv.org/abs/2605.03677.

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggyu Lim, and Taesup Kim. Stable On-Policy Distillation through Adaptive Target Reformulation. *arXiv preprint arXiv:2601.07155*, 2026. URL https://arxiv.org/abs/2601.07155.

Li Jiang, Haoran Xu, Yichuan Ding, and Amy Zhang. Trajectory-refined distillation. *arXiv preprint arXiv:2606.08432*, 2026a. URL https://arxiv.org/abs/2606.08432.

Yinuo Jiang, Yongjie Ye, Zhou Tao, Xiang Zhuang, Qiang Zhang, Huajun Chen, and Tiankai Li. When Teachers Mislead: Spurious-Signal-Aware On-Policy Distillation. *arXiv preprint arXiv:2608.03632*, 2026b. URL https://arxiv.org/abs/2608.03632.

Yuxuan Jiang and Francis Ferraro. Bridging reasoning trajectories in on-policy distillation via near-future guidance. *arXiv preprint arXiv:2606.00305*, 2026. URL https://arxiv.org/abs/2606.00305.

Yuxuan Jiang, Runchao Li, Shubhashis Roy Dipta, Dawei Li, and Zhao Yang. Cornerstones or Stumbling Blocks? Deciphering the Rock Tokens in On-Policy Distillation. *arXiv preprint arXiv:2605.09253*, 2026c. URL https://arxiv.org/abs/2605.09253.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.06770.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-Aware On-Policy Distillation of Language Models. *arXiv preprint arXiv:2603.07079*, 2026. URL https://arxiv.org/abs/2603.07079.

Wonjun Kang, Kevin Galim, Seunghyuk Oh, Minjun Kang, Sanghyun Park, Donghoon Kim, Minjae Lee, Minseo Kim, Rishabh Tiwari, Yuchen Zeng, Hyung Il Koo, and Kangwook Lee. AsyncOPD: How Stale Can On-Policy Distillation Be? *arXiv preprint arXiv:2606.24143*, 2026. URL https://arxiv.org/abs/2606.24143.

Junlong Ke, Zichen Wen, Weijia Li, Conghui He, and Linfeng Zhang. Respecting Self-Uncertainty in On-Policy Self-Distillation for Efficient LLM Reasoning. *arXiv preprint arXiv:2605.13255*, 2026. URL https://arxiv.org/abs/2605.13255.

Gyeongman Kim, Doohyuk Jang, and Eunho Yang. PromptKD: Distilling Student-Friendly Knowledge for Generative Language Models via Prompt Tuning. *arXiv preprint arXiv:2402.12842*, 2024. URL https://arxiv.org/abs/2402.12842.

Jeonghye Kim, Xufang Luo, Minbeom Kim, Sangmook Lee, Dohyung Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Why Does Self-Distillation (Sometimes) Degrade the Reasoning Capability of LLMs? *arXiv preprint arXiv:2603.24472*, 2026. URL https://arxiv.org/abs/2603.24472.

Yeongmin Kim, Donghyeok Shin, Mina Kang, Byeonghu Na, and Il-Chul Moon. Distillation of Large Language Models via Concrete Score Matching. *arXiv preprint arXiv:2509.25837*, 2025. URL https://arxiv.org/abs/2509.25837.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, 2016. doi: 10.18653/v1/D16-1139.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2402.03898.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2503.07067.

Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. Scaling Reasoning Efficiently via Relaxed On-Policy Distillation. *arXiv preprint arXiv:2603.11137*, 2026. URL https://arxiv.org/abs/2603.11137.

Xin Li, Hao Jiang, Annan Wang, Yichi Zhang, and Chau Yuen. The Extrapolation Cliff in On-Policy Distillation of Near-Deterministic Structured Outputs. *arXiv preprint arXiv:2605.08737*, 2026a. URL https://arxiv.org/abs/2605.08737.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking On-Policy Distillation of Large Language Models: Phenomenology, Mechanism, and Recipe. *arXiv preprint arXiv:2604.13016*, 2026b. URL https://arxiv.org/abs/2604.13016.

Zhaoyi Li, Deyang Kong, Yuan Wei, Evan Yang, Ranran Shen, Mahardika Krisna Ihsani, Ming Yang, Wei Zhang, Chuan Hao, Jian Yang, Ran Tao, Bryan Dai, Shikun Zhang, Wei Ye, Ying Wei, and Defu Lian. Every Coin Has Two Sides: On the Dual Nature of Generalization in On-Policy Distillation of Large Language Models. *arXiv preprint arXiv:2608.16647*, 2026c. URL https://arxiv.org/abs/2608.16647.

Junwei Liu, Kaixin Wang, Yixuan Chen, Xin Peng, Zhenpeng Chen, Lingming Zhang, and Yiling Lou. Large language model-based agents for software engineering: A survey. *arXiv preprint arXiv:2409.02977*, 2024. URL https://arxiv.org/abs/2409.02977.

Kaiyuan Liu, Ziyuan Zhuang, Yang Bai, Bing Wang, Rongxiang Weng, and Jieping Ye. Prefix Teach, Suffix Fade: Local Teachability Collapse in Strong-to-Weak On-Policy Distillation. *arXiv preprint arXiv:2605.13643*, 2026a. URL https://arxiv.org/abs/2605.13643.

Tao Liu, Taiqiang Wu, Mao Zheng, Xuan Luo, Runming Yang, Xuewei Yang, Junjie Wang, and Yujiu Yang. Simple-OPD: Demystifying Warm-up for On-policy Distillation. *arXiv preprint arXiv:2608.06802*, 2026b. URL https://arxiv.org/abs/2608.06802.

Xiaogeng Liu, Xinyan Wang, Yingzi Ma, Yechao Zhang, and Chaowei Xiao. When Are Teacher Tokens Reliable? Position-Weighted On-Policy Self-Distillation for Reasoning. *arXiv preprint arXiv:2605.21606*, 2026c. URL https://arxiv.org/abs/2605.21606.

Xinyu Liu, Kechen Jiao, Chunyang Xiao, Runsong Zhao, Junhao Ruan, Bei Li, Jiahao Liu, Qifan Wang, Xin Chen, Jingang Wang, Chenglong Wang, Tong Xiao, and JingBo Zhu. Teacher-Guided Policy Optimization for On-Policy Reasoning Distillation under Large Policy Divergence. *arXiv preprint arXiv:2605.13230*, 2026d. URL https://arxiv.org/abs/2605.13230.

Yanjiang Liu, Jie Lou, Xinyan Guan, Yuqiu Ji, Hongyu Lin, Ben He, Xianpei Han, Le Sun, Xing Yu, and Yaojie Lu. Your Teacher Can't Help You Here: Combating Supervision Fidelity Decay in On-Policy Distillation. *arXiv preprint arXiv:2605.30833*, 2026e. URL https://arxiv.org/abs/2605.30833.

Feng Luo, Yu-Neng Chuang, Guanchu Wang, Zicheng Xu, Xiaotian Han, Tianyi Zhang, and Vladimir Braverman. Demystifying OPD: Length Inflation and Stabilization Strategies for Large Language Models. *arXiv preprint arXiv:2604.08527*, 2026. URL https://arxiv.org/abs/2604.08527.

OpenAI. Introducing SWE-bench Verified. https://openai.com/index/introducing-swe-bench-verified/, 2024. Accessed 2026-08-20.

Zikun Qu, Min Zhang, Mingze Kong, Zhiwei Shang, Zhengyu Chen, Yikun Ban, Shuang Qiu, and Zhongxiang Dai. SPOT: Sparse Probing and Outcome Calibration for On-Policy Distillation. *arXiv preprint arXiv:2608.04419*, 2026. URL https://arxiv.org/abs/2608.04419.

Qwen Team. Qwen3-Coder: Agentic coding in the world. https://qwenlm.github.io/blog/qwen3-coder/, 2025. Accessed 2026-08-20.

Miao Rang, Zhenni Bi, Hang Zhou, Kai Han, Xuechun Wang, An Xiao, Xinghao Chen, Yunhe Wang, and Hanting Chen. Near-Policy: Accelerating On-Policy Distillation via Asynchronous Generation and Selective Packing. *arXiv preprint arXiv:2605.05940*, 2026. URL https://arxiv.org/abs/2605.05940.

Scale Labs. SWE Atlas: Benchmarking coding agents beyond issue resolution. https://labs.scale.com/papers/sweatlas, 2026. Accessed 2026-08-20.

Zhennan Shen, Yanshu Li, Qingyu Yin, Chak Tou Leong, Zhilin Wang, Yanxu Chen, Rongduo Han, Sunbowen Lee, and Yi R. Fung. On the Geometry of On-Policy Distillation. *arXiv preprint arXiv:2606.07082*, 2026. URL https://arxiv.org/abs/2606.07082.

Daniil Tiapkin, Daniele Calandriello, Johan Ferret, Sarah Perrin, Nino Vieillard, Alexandre Ramé, and Mathieu Blondel. On Teacher Hacking in Language Model Distillation. *arXiv preprint arXiv:2502.02671*, 2025. URL https://arxiv.org/abs/2502.02671.

Jiaqi Wang, Wenhao Zhang, Weijie Shi, Yaliang Li, and James Cheng. TCOD: Exploring Temporal Curriculum in On-Policy Distillation for Multi-turn Autonomous Agents. *arXiv preprint arXiv:2604.24005*, 2026a. URL https://arxiv.org/abs/2604.24005.

Jiaxuan Wang, Xuan Ouyang, Zhiyu Chen, Yulan Hu, Zheng Pan, Xin Li, and Lan-Zhe Guo. TRACE: Distilling Where It Matters via Token-Routed Self On-Policy Alignment. *arXiv preprint arXiv:2605.10194*, 2026b. URL https://arxiv.org/abs/2605.10194.

Rui Wang, Hongru Wang, Yi Chen, Boyang Xue, Tianqing Fang, Wenhao Yu, and Kam-Fai Wong. Demystifying On-Policy Distillation: Roles, Pathologies, and Regulations. *arXiv preprint arXiv:2607.13399*, 2026c. URL https://arxiv.org/abs/2607.13399.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, et al. OpenHands: An open platform for AI software developers as generalist agents. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2407.16741.

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

Yecheng Wu, Song Han, and Han Cai. Lightning OPD 2.0: Mitigating Style Bias in Cross-Teacher On-Policy Distillation for Large Reasoning Models. *arXiv preprint arXiv:2607.28449*, 2026. URL https://arxiv.org/abs/2607.28449.

Jinwei Xiao, Zhuowen Han, Yueqing Sun, Zhengxi Lu, Yuxin Liu, Zhiyuan Yao, Wentao Chen, Qi Gu, and Xunliang Cai. Finding the Evidence: Discovering Decision-Supporting Tokens for On-Policy Reasoning Distillation. *arXiv preprint arXiv:2606.22830*, 2026. URL https://arxiv.org/abs/2606.22830.

Yan Xie, Sijie Zhu, Tiansheng Wen, Bo Chen, and Yifei Wang. On the Position Bias of On-Policy Distillation. *arXiv preprint arXiv:2606.22600*, 2026. URL https://arxiv.org/abs/2606.22600.

Xingrun Xing, Haoqing Wang, Boyan Gao, Ziheng Li, and Yehui Tang. Trust Region On-Policy Distillation. *arXiv preprint arXiv:2606.01249*, 2026. URL https://arxiv.org/abs/2606.01249.

Haolei Xu, Xiaowen Xu, Haiwen Hong, Zixuan Ni, Hongxing Li, Yiwen Qiu, Weiming Lu, and Yongliang Shen. Pass the baton: Trajectory-relayed on-policy distillation. *arXiv preprint arXiv:2607.26057*, 2026a. URL https://arxiv.org/abs/2607.26057.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, and Zhipeng Wang. PACED: Distillation and On-Policy Self-Distillation at the Frontier of Student Competence. *arXiv preprint arXiv:2603.11178*, 2026b. URL https://arxiv.org/abs/2603.11178.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. TIP: Token Importance in On-Policy Distillation. *arXiv preprint arXiv:2604.14084*, 2026c. URL https://arxiv.org/abs/2604.14084.

Fanjia Yan. A function calling perspective on scalable large language model agent evaluation. Technical Report UCB/EECS-2025-184, University of California, Berkeley, 2025. URL https://www2.eecs.berkeley.edu/Pubs/TechRpts/2025/EECS-2025-184.html.

Chen Yang, Haiyuan Wan, Rengrong Xiong, Yize Chen, and Danny H.K. Tsang. Beyond Imitation: Filtering On-Policy Distillation by Reasoning Progress. *arXiv preprint arXiv:2608.19408*, 2026a. URL https://arxiv.org/abs/2608.19408.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. *arXiv preprint arXiv:2405.15793*, 2024. URL https://arxiv.org/abs/2405.15793.

John Yang, Kilian Lieret, Carlos E. Jimenez, Alexander Wettig, Kabir Khandpur, Yanzhe Zhang, Binyuan Hui, Ofir Press, Ludwig Schmidt, and Diyi Yang. Swe-smith: Scaling data for software engineering agents, 2025. URL https://arxiv.org/abs/2504.21798.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond Teacher: Generalized On-Policy Distillation with Reward Extrapolation. *arXiv preprint arXiv:2602.12125*, 2026b. URL https://arxiv.org/abs/2602.12125.

Zhicheng Yang, Zhijiang Guo, Yifan Song, Minrui Xu, Yongxin Wang, Yiwei Wang, Xiaodan Liang, and Jing Tang. Prune-OPD: Efficient and Reliable On-Policy Distillation for Long-Horizon Reasoning. *arXiv preprint arXiv:2605.07804*, 2026c. URL https://arxiv.org/abs/2605.07804.

Guo Yu, Wenlin Liu, Yulan Hu, Hao-Xuan Ma, Jun-Peng Jiang, and Han-Jia Ye. Dense Supervision, Sparse Updates: On the Sparsity and Geometry of On-Policy Distillation. *arXiv preprint arXiv:2606.13657*, 2026a. URL https://arxiv.org/abs/2606.13657.

Zichao Yu, Chengzhi Yu, Shengze Xu, Yujin Han, Bingqing Jiang, Xu Wang, and Difan Zou. Mismatch Matters: On-Policy Distillation Beyond Token Agreement. *arXiv preprint arXiv:2608.09836*, 2026b. URL https://arxiv.org/abs/2608.09836.

Dongxu Zhang, Zhichao Yang, Sepehr Janghorbani, Jun Han, Andrew Ressler II, Qian Qian, Gregory D. Lyng, Sanjit Singh Batra, and Robert E. Tillman. Fast and Effective On-policy Distillation from Reasoning Prefixes. *arXiv preprint arXiv:2602.15260*, 2026a. URL https://arxiv.org/abs/2602.15260.

Jiaxin Zhang, Xiangyu Peng, Qinglin Chen, Qinyuan Ye, Caiming Xiong, and Chien-Sheng Wu. The Illusion of Certainty: Decoupling Capability and Calibration in On-Policy Distillation. *arXiv preprint arXiv:2604.16830*, 2026b. URL https://arxiv.org/abs/2604.16830.

Zhiwei Zhang, Zechen Sun, Fei Zhao, Kang Peng, Bin Liang, Huayu Deng, Yao Hu, Kam-Fai Wong, and Mu Chuan. Verify Before You Distill: Prompt-Level Teacher Gating for On-Policy Distillation. *arXiv preprint arXiv:2609.02998*, 2026c. URL https://arxiv.org/abs/2609.02998.

Anhao Zhao, Junlong Tong, Yingqi Fan, Ping Nie, Wenjie Li, and Xiaoyu Shen. PowerOPD: Stabilizing On-Policy Distillation with Bounded Power Transformation. *arXiv preprint arXiv:2606.17199*, 2026a. URL https://arxiv.org/abs/2606.17199.

Hanyang Zhao, Haoxian Chen, Han Lin, Genta Indra Winata, David Yao, and Wenpin Tang. OPD+: Rethinking the Advantage Design for On-Policy Distillation. *arXiv preprint arXiv:2606.01039*, 2026b. URL https://arxiv.org/abs/2606.01039.

Binbin Zheng, Xing Ma, Yiheng Liang, Jingqing Ruan, Xiaoliang Fu, Kepeng Lin, Benchang Zhu, Ke Zeng, and Xunliang Cai. SCOPE: Signal-Calibrated On-Policy Distillation Enhancement with Dual-Path Adaptive Weighting. *arXiv preprint arXiv:2604.10688*, 2026. URL https://arxiv.org/abs/2604.10688.

Qiyong Zhong, Mao Zheng, Mingyang Song, Xin Lin, Jie Sun, Houcheng Jiang, Xiang Wang, and Junfeng Fang. SOD: Step-wise on-policy distillation for small language model agents. *arXiv preprint arXiv:2605.07725*, 2026. URL https://arxiv.org/abs/2605.07725.

Yuhang Zhou, Kai Zheng, Haoling Li, Dengyun Peng, Can Xu, and Jingjing Chen. TurnOPD: Making on-policy distillation turn-aware for efficient long-horizon agent training. *arXiv preprint arXiv:2607.05804*, 2026. URL https://arxiv.org/abs/2607.05804.

Siqi Zhu, Xuyan Ye, Hongyu Lu, Weiye Shi, and Ge Liu. The Many Faces of On-Policy Distillation: Pitfalls, Mechanisms, and Fixes. *arXiv preprint arXiv:2605.11182*, 2026a. URL https://arxiv.org/abs/2605.11182.

Wenhong Zhu, Ruobing Xie, Rui Wang, and Pengfei Liu. Hybrid Policy Distillation for LLMs. *arXiv preprint arXiv:2604.20244*, 2026b. URL https://arxiv.org/abs/2604.20244.

Ximo Zhu, Ruiqi Liu, Rong Wang, Ping Wu, Xiang Zheng, Wenzhuo Xu, Xubin Yao, Zhiyuan Yan, Bo Li, Jun Gao, and Xiaolei Lv. ReOrder-OPD:Reliability-Aware Prompt Ordering for On-Policy Distillation. *arXiv preprint arXiv:2608.10905*, 2026c. URL https://arxiv.org/abs/2608.10905.

Zhou Ziheng, Jiaqi Li, Huacong Tang, Ying Nian Wu, and Demetri Terzopoulos. Less is More: Early Stopping Rollout for On-Policy Distillation. *arXiv preprint arXiv:2605.27028*, 2026. URL https://arxiv.org/abs/2605.27028.